CHAPTER TWO

IMAGE PROCESSING AND COMPUTER VISION

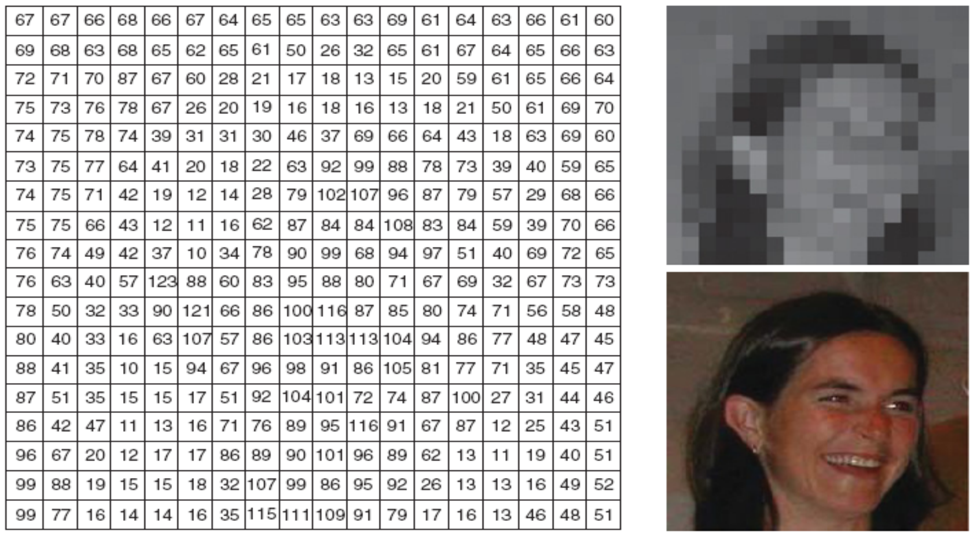
Image Processing and Computer Vision

1 Introduction [01]:

Computer vision is the automatic analysis of images and videos by computers in order togain some understanding of the world. Computer vision is inspired by the capabilities ofthe human vision system and, when initially addressed in the 1960s and 1970s, it was thought tobe a relatively straightforward problem to solve. However, the reason we think/thought thatvision is easy is that we have our own visual system which makes the task seem intuitive toour conscious minds. In fact, the human visual system is very complex and evens the estimatesof how much of the brain is involved with visual processing vary from 25% up to morethan 50%.

1.1 A Difficult Problem:

The first challenge facing anyone studying this subject is to convince them-self that the problemis difficult. To try to illustrate the difficulty, we first show three different versions of the same image in Figure 1.1. For a computer, an image is just an array of values, such as the array shown in the left-hand image in Figure 1.1. For us, using our complex vision system, we canperceive this as a face image but only if we are shown it as a grey scale image (top right).Computer vision is quite like understanding the array of values shown in Figure 1.1, but is more complicated as the array is really much bigger (e.g. to be equivalent to the human eyea camera would need around 127 million elements), and more complex (i.e. with each point represented by three values in order to encode color information).



67	67	66	68	66	67	64	65	65	63	63	69	61	64	63	66	61	60
69	68	63	68	65	62	65	61	50	26	32	65	61	67	64	65	66	63
72	71	70	87	67	60	28	21	17	18	13	15	20	59	61	65	66	64
75	73	76	78	67	26	20	19	16	18	16	13	18	21	50	61	69	70
74	75	78	74	39	31	31	30	46	37	69	66	64	43	18	63	69	60
73	75	77	64	41	20	18	22	63	92	99	88	78	73	39	40	59	65
74	75	71	42	19	12	14	28	79	102	107	96	87	79	57	29	68	66
75	75	66	43	12	11	16	62	87	84	84	108	83	84	59	39	70	66
76	74	49	42	37	10	34	78	90	99	68	94	97	51	40	69	72	65
76	63	40	57	123	88	60	83	95	88	80	71	67	69	32	67	73	73
78	50	32	33	90	121	66	86	100	116	87	85	80	74	71	56	58	48
80	40	33	16	63	107	57	86	103	113	113	104	94	86	77	48	47	45
88	41	35	10	15	94	67	96	98	91	86	105	81	77	71	35	45	47
87	51	35	15	15	17	51	92	104	101	72	74	87	100	27	31	44	46
86	42	47	11	13	16	71	76	89	95	116	91	67	87	12	25	43	51
96	67	20	12	17	17	86	89	90	101	96	89	62	13	11	19	40	51
99	88	19	15	15	18	32	107	99	86	95	92	26	13	13	16	49	52
99	77	16	14	14	16	35	115	111	109	91	79	17	16	13	46	48	51

Figure 1.1 Different versions of an image. An array of numbers (left) which are the values of the grey scales in the low resolution image of a face (top right). The task of computer vision is most like understanding the array of numbers

[Adopted][01]

1.2 Practical Applications of Computer Vision[01]

Computer vision has many applications in industry, particularly allowing the automatic inspectionof manufactured goods at any stage in the production line. For example, it has been used to:

- Inspect printed circuits boards to ensure that tracks and components are placed correctly .See Figure 1.2.
- Inspect print quality of labels. See Figure 1.3.
- Inspect bottles to ensure they are properly filled. See Figure 1.3.

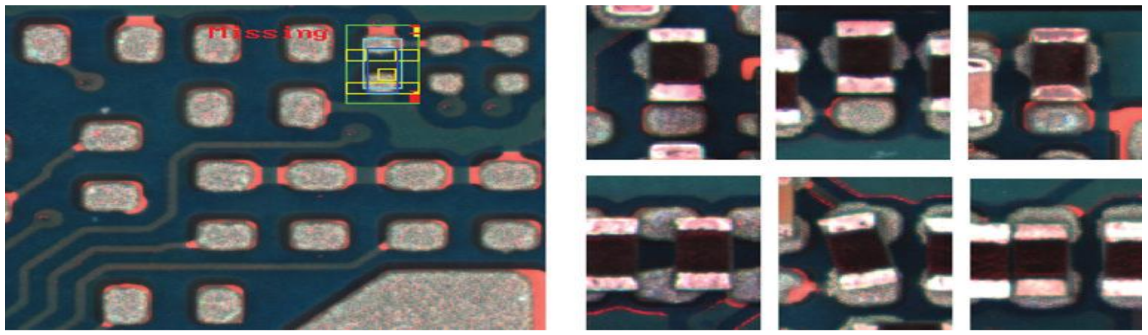


Figure 1.2 PCB inspection of pads (left) and images of some detected flaws in the surface mounting of components (right). Reproduced by permission of James Mahon [adopted][01]

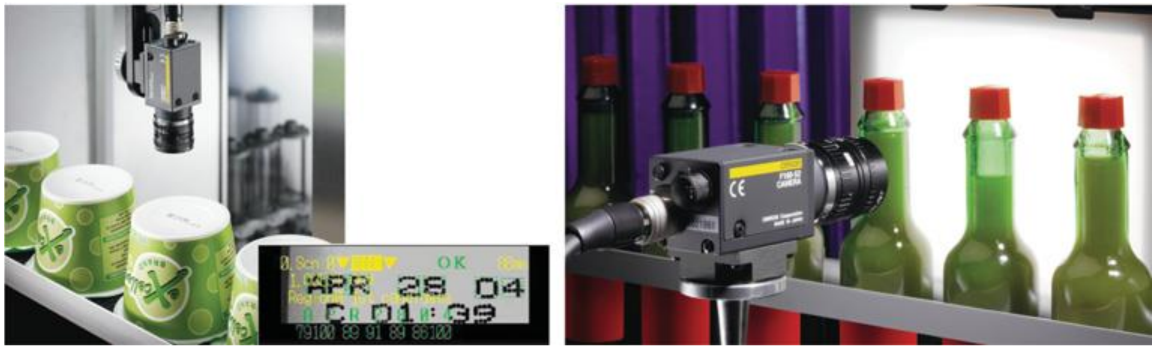


Figure 1.3 Checking print quality of best-before dates (right), and monitoring level to which bottles are filled (right). Reproduced by permission of Omron Electronics LLC[adopted][01]

- Inspect apples to determine if there is any bruising.
- Locate chocolates on a production line so that a robot arm can pick them up and place themin the correct locations in the box.
- Guide robots when manufacturing complex products such as cars. On the factory floor, the problem is a little simpler than in the real world as the lightingcan be constrained and the possible variations of what we can see are quite limited. Computervision is now solving problems outside the factory. Computer vision applications outside the

<u>Factory includes:</u>

- The automatic reading of license plates as they pass through tollgates on major roads.
- Augmenting sports broadcasts by determining distances for penalties, along with range ofother statistics (such as how far each player has travelled during the game).
- Biometric security checks in airports using images of faces and images of fingerprints. See Figure 1.4.
- Augmenting movies by the insertion of virtual objects into video sequences, so that theyappear as though they belong (e.g. the candles in the Great Hall in the Harry Potter movies).

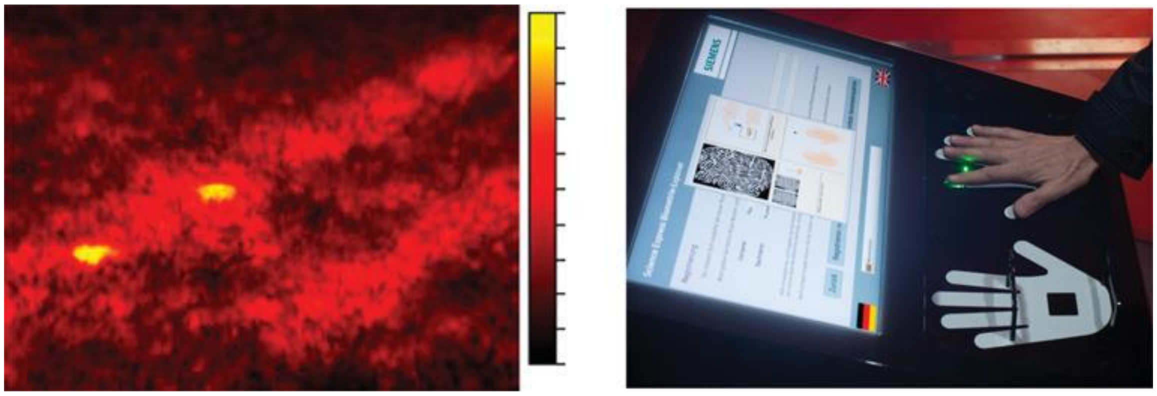


Figure 1.4 Buried landmines in an infrared image (left). Reproduced by permission of ZouheirFawaz, Handprint recognition system (right). Reproduced by permission of Siemens AG[adopted][01]

- Assisting drivers by warning them when they are drifting out of lane.
- Creating 3D models of a destroyed building from multiple old photographs.

- Advanced interfaces for computer games allowing the real time detection of players or theirhand-held controllers.
- Classification of plant types and anticipated yields based on multispectral satellite images.
- Detecting buried landmines in infrared images. See Figure 1.4.

1.3 The Future of Computer Vision

The community of vision developers is constantly pushing the boundaries of what we canachieve. While we can produce autonomous vehicles, which drive themselves on a highway,we would have difficulties producing a reliable vehicle to work on minor roads, particularly ifthe road marking were poor. Even in the highway environment, though, we have a legal issue,as who is to blame if the vehicle crashes? Clearly, those developing the technology do not thinkit should be them and would rather that the driver should still be responsible should anythinggo wrong. This issue of liability is a difficult one and arises with many vision applications inthe real world. Taking another example, if we develop a medical imaging system to diagnosecancer, what will happen when it mistakenly does not diagnose a condition? Even thoughthe system might be more reliable than any individual radiologist, we enter a legal minefield.Therefore, for now, the simplest solution is either to address only non-critical problems orto develop systems, which are assistants to, rather than replacements for, the current humanexperts.Another problem exists with the deployment of computer vision systems. In some countriesthe installation and use of video cameras is considered an infringement of our basic right toprivacy. This varies hugely from country to country, from company to company, and evenfrom individual to individual. While most people involved with technology see the potentialbenefits of camera systems, many people are inherently distrustful of video cameras and whatthe videos could be used for. Among other things, they fear (perhaps justifiably) a Big Brotherscenario, where our movements and actions are constantly monitored. Despite this, the numberof cameras is growing very rapidly, as there are cameras on virtually every new computer,every new phone, every new games console, and so on.Moving forwards, we expect to see computer vision addressing progressively harder problems;that is problems in more complex environments with fewer constraints. We expectcomputer vision to start to be able to recognise more objects of different types and to begin toextract more reliable and robust descriptions of the world in which they operate. For example,
we expect computer vision to

- become an integral part of general computer interfaces;
- provide increased levels of security through biometric analysis;
- provide reliable diagnoses of medical conditions from medical images and medical records;
- allow vehicles to be driven autonomously;
- Automatically determine the identity of criminals through the forensic analysis of video.

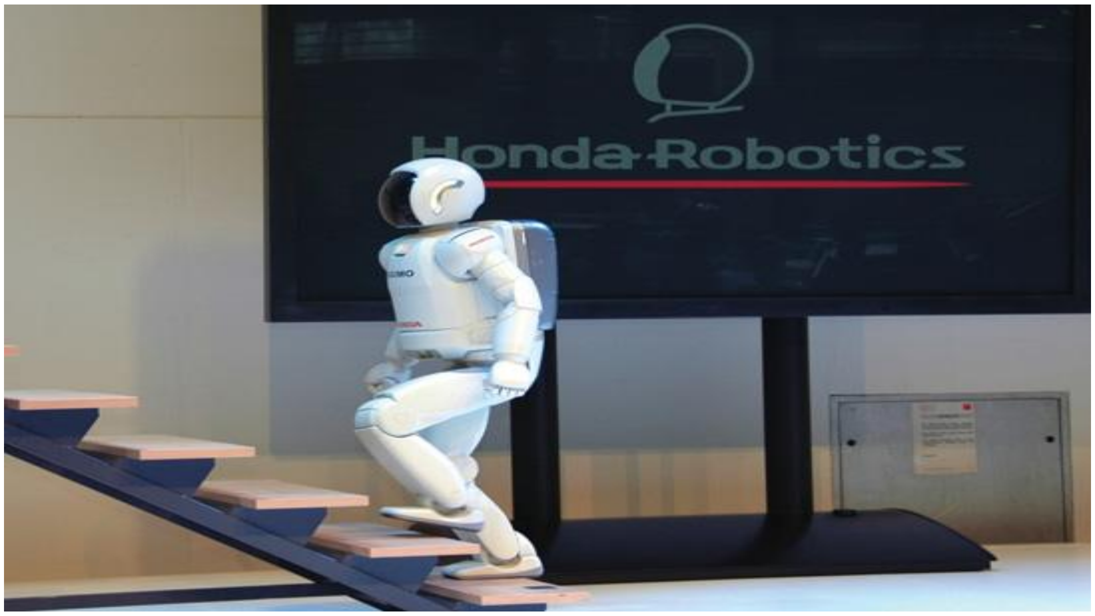


Figure 1.5 The ASIMO humanoid robot which has two cameras in its 'head' which allow ASIMO to determine how far away things are, recognise familiar faces, etc. Reproduced by permission of Honda Motor Co. Inc[adopted][01]

Ultimately, computer vision is aiming to emulate the capabilities of human vision,
And Toprovide these abilities to humanoid (and other) robotic devices, such as ASIMO (See Figure1.5). This is part of what makes this field exciting, and surprising, as we all have our own(human) vision systems which work remarkably well, yet when we try to automate anycomputer vision task it proves very difficult to do reliably.

1.4 Computer vision tools [02]

- VXL is a set of C++ computer vision libraries with support for imaging and streaming IO, geometry for advanced graphics and generic number-crunching algorithms for matrix or vector manipulation (See link). You also have a fair level of GUI development support which can be checked at their online book at the following link. (http://public.kitware.com/vxl/doc/release/books/core/book_10.html#SEC112).
- BoofCV: stereo ... features in pure java !
  the list of features designed for version 1.0 :
  (http://boofcv.org/index.php?title=BoofCV_Roadma)
- OpenCV with Python :
  (http://opencvpython.blogspot.jp/)

AForge.NET is a C# framework designed for researchers in the fields of Computer Vision and Artificial Intelligence: image processing, computer vision, neural networks, genetic algorithms, machine learning.

CHAPTER TWO

Image Processing and Computer Vision

Note :-
Will using the open-cv to implement the computer vision.

2 Images [01]:

Images are central to computer vision, as they are the representation that we obtain from Imaging devices .They provides us with a representation of the visual appearance of a scene which we can process to enhance certain features of interest, before we attempt to abstract information. Images generally exhibit some degree of noise, which can be attenuated by various simple image processing techniques.

2.1 Cameras:

A camera consists of a photosensitive image plane (which senses the amount of light that falls upon it), a housing which prevents stray light from falling onto the image plane and a lens inthe housing which allows some light to fall onto the image plane in a controlled fashion (i.e. the light rays are focused onto the image plane by the lenses).

2.2 Images

An image is a picture (generally a 2D projection of a 3D scene) captured by a sensor. It is A continuous function of two coordinates in the image plane; usually expressed as (i,j) or (Column, row) or somewhat confusingly (x,y). To process such an image on a digital com-puter.It must be both

- Sampled into a matrix (M rows and N columns), and
- Quantized so that each element in the matrix is given an integer value. In other words, the continuous range is split into some number of intervals (k) where most commonly k = 256.(figure 2.1)

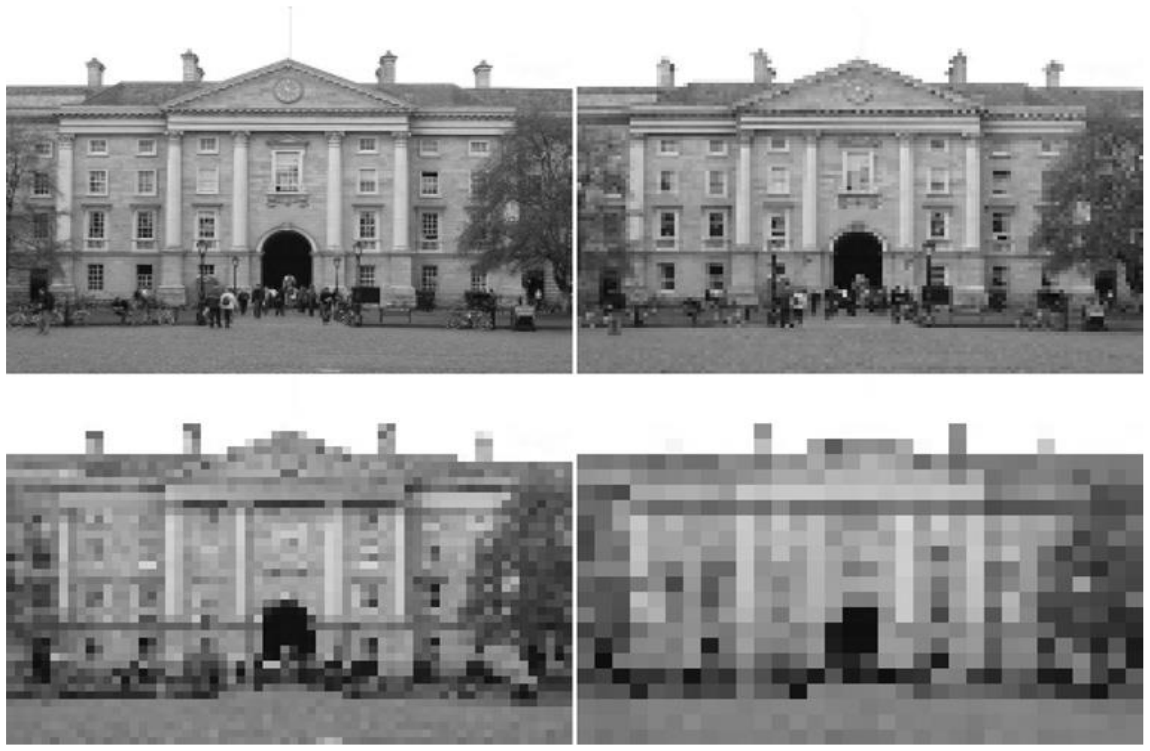


Figure 2.1 Four different samplings of the same image; top left 256x192, top right 128x96, bottom left 64x48 and bottom right 32x24[adopted][01].

2.3 Color Images:

Color (multispectral) images (Plataniotis&Venetsanopoulos, 2000) (Gevers, Gijsenij, van de Weijer, &Geusebroek, 2012) have multiple channels, whereas grey-scale (monochromatic) Images (sometimes, incorrectly, referred to as black and white images) have only one channel. A grey-scale image represents the luminance (Y) at every point a scene. A color image represents both luminance and chrominance (color information) within the scene. This informationcan be represented in a number of ways but in all cases requires multiple channels ofimage data. Hence, color images (with the same sampling and quantization) are bigger andmore complex than grey-scale images, as we must somehow decide how to process each ofthe channels of information. Note that much of image processing was developed specificallyfor grey-scale images and its application to color images is often not very well defined.Computer vision was for many years based on grey-level images, mainly based on twopremises:

- Humans can understand grey-level images, so why bother with color?
- Grey-scale images are smaller and less complex (a number for each point).

However, colour does provide more useful information that can assist with many tasks, such as segmentation of the image into physically separate entities (objects, surfaces, etc.). For example, looking at Figure 2.1 you should find it much easier to separate/segment the different trees in the color image.

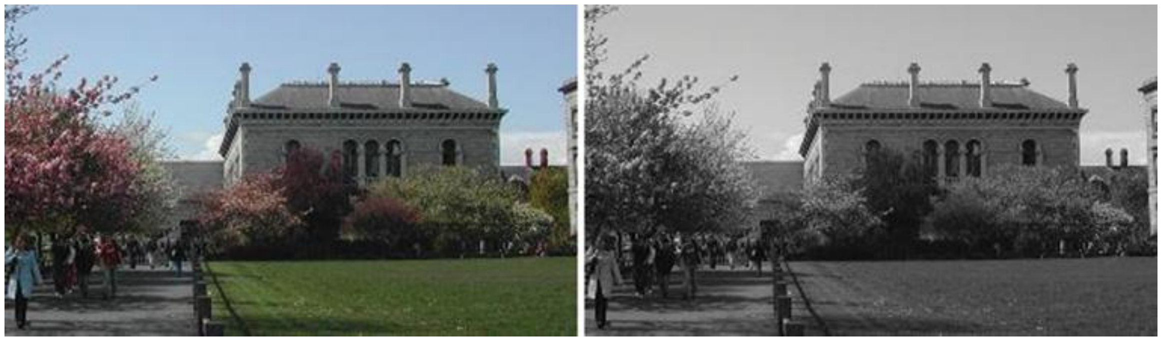


Figure 2.2 RGB color image (left) and the same image in grey-scale (right)[adopted][01]

Humans are sensitive to light at wavelengths between 400 nm and 700 nm and hence most camera image sensors are designed to be sensitive at those wavelengths.
Color images are more complex than grey-scale images and are typically represented using a three-channel color space (Koschan&Abidi, 2007).

[01]*A Practical Introduction to Computer Vision with OpenCV*, First Edition. Kenneth Dawson-Howe.
© 2014 John Wiley & Sons, Ltd. Published 2014 by John Wiley & Sons, Ltd.
[02]https://www.researchgate.net/post/Which_are_the_best_open_source_tools_for_image_processing_and_computer_vision

Voice-Recognition

3 Introduction:

[01]The theme of Social interaction and intelligence is important and interesting to an Artificialintelligence and Robotics community. It is one of the challenging areas inHuman-Robot Interaction (HRI). Speech recognition technology is a great aid to admitthe challenge and it is a prominent technology for Human-Computer Interaction (HCI)and Human-Robot Interaction (HRI) for the future.Humans are used to interact with Natural Language (NL) in the social context. Thisidea leads Robotics to make NL interface through Speech for the HRI. Natural Language(NL) interface is now starting to appear in standard software application. Thisgives benefit to novices to easily interact with the standard software in HCI field. It'salso encouraging Robotics to use Speech Recognition (SR) technology for the HRI. Topercept the world is important knowledge for the knowledge Based-Agent and Robot todo a task. It's also a key factor to know initial knowledge about the Unknown world.In the social context Robot can easily interact with Human through SR to gain theinitial knowledge about the Unknown world and also the information about the task to Accomplish. There are several SR interface robotic systems have been presented. Most of the projects emphasize on Mobile Robot - now a days this type of robot is gettingpopular as a service robot at indoor and outdoor1. The goal of the service robot isto help people in everyday life at social context. It is an important thing for the Mobile robot to communicate with the users (human) of its world. So Speech Recognition (SR)is an easy way of communication with human and it also gives the advantage of interactingwith the novice users without a proper training. Uncertainty is a major problemfor navigation systems in mobile robots - interaction with humans in a natural way,using English rather than a programming language, would be a means of overcomingdifficulties with localization.

3.1 Speech Recognition

Speech Recognition technology promises to change the way we interact with machines (robots, computers etc.) in the future. This technology is getting matured day by day and scientists are still working hard to overcome the remaining limitation. Now a days it is introducing many important areas (like - in the field of Aerospace where the training and operational demands on the crew have significantly increased with the proliferation of technology, in the Operation Theater as a surgeon's aid to control lights, cameras, pumps and equipment by simple voice commands) in the social context. Speech recognition is the process of converting an acoustic signal, captured by microphone or a telephone, to a set of words .There two important part of in Speech Recognition

- Recognize the series of sound
- Identified the word from the sound. This recognition technique depends also on many parameters - Speaking Mode, Speaking Style, Speaker Enrollment, Size of the Vocabulary, Language Model, Perplexity, Transducer etc.

There are two types of Speak Mode for speech recognition system:

- one word at a time (isolated-word speech)
- Continuous speech.

Depending on the speaker enrolment, the speech recognition system can also divide - Speaker dependent and Speaker independent system. In Speaker dependent systems user need to be train the systems before using them, on the other hand Speaker independent system can identify any speaker's speech.

3.2 VUI (Voice user interface)

In Robotics User interface is an important component of any product handle by the human user. The concept of robotics is to make an autonomous machine, which can replace human labor. But, to control the robot or to provide guide line for work, human should communicate with the robot and this concept conclude the Robotics to introduce User Interface to communicate with robot. In the past decades GUI (Graphical User Interface), Keyboard, Keypad, Joystick is the dominating tools for Interaction with machine. Now there are several new technologies are introducing in Human machine interaction filed; from them SR system is one of the interesting tool to the researchers for interaction with machine. The reason - it (SR system) draw attention to the researcher, because people are used to communicate with Natural Language (NL) in the social context; so this technology can be widely-accepted to the human user fairly and easily. The Robotics is also getting interest in SR system or VUI (Voice User Interface) for the same reason. With the addition of Hearing Sensor (SR system), the concept of humanoid robot also becomes true. Show figure3.1.

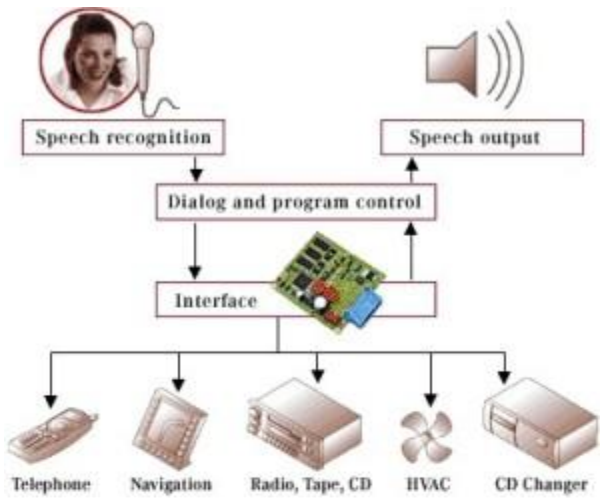


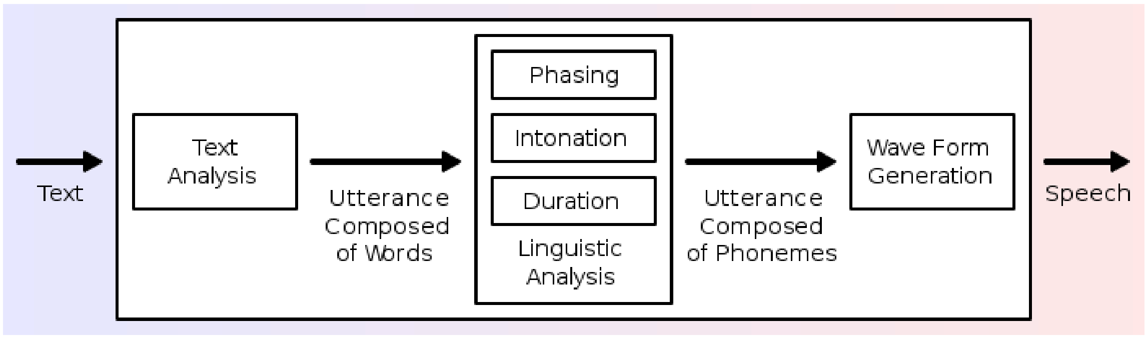
Figure 3.1: A Broadly defined SR System [adopted][01]

3.2.1 STT (Speech-To-Text)

A**Speech-To-Text (STT)** system converts text into normal language.

3.2.2 TTS (Text-To-Speech)

A **text-to-speech** (**TTS**) system converts normal language text into speech. Show figure3.2



[Figure 3.2]Overview of a typical TTS system[adopted][01]

References

[01] https://en.wikipedia.org/wiki/Speech_synthesis
[02] http://ecurrent.fit.edu/blog/panther-voices/career-possibilities-speech-recognition-engineer/#prettyPhoto

Glossary

GUI - Graphical User Interface
HCI - Human-Computer Interaction
HRI - Human-Robot Interaction
NL - Natural Language
SR - Speech Recognition
VUI - Voice User Interface
UI - User Interface
TTS - Text-To-Speech

Single Board Computing

4 Introduction

With the advent of technology in the consumer electronics domain, single board computers have become quite popular among both consumers and developers. These days everyone has virtually become so much “wired” that they cannot live without these so called gadgets. Right from the mobile phone in your pockets to high end gaming consoles, including tablets, PCs, iPod, etc., everything is basically a single board computer.

4.1 Single Board Computers

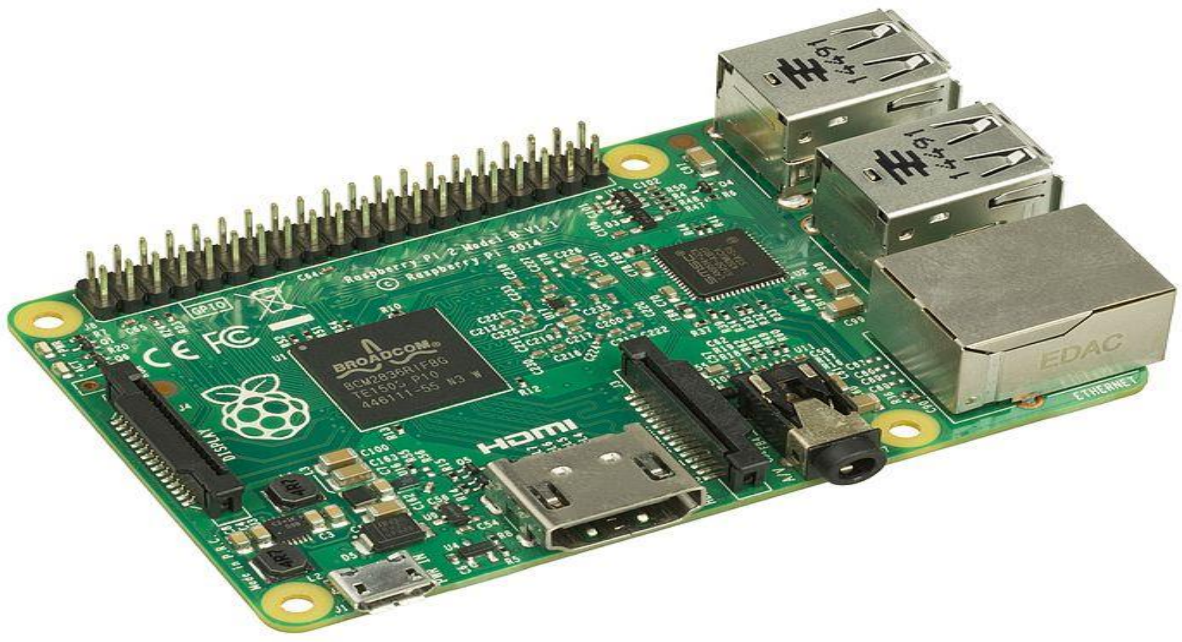
There is a difference between traditional computers and single board computers. You must be familiar that full-fledged computers (like PCs and Mac) have a motherboard. On the motherboard, you will essentially find a processor (like the Intel® Core™, AMD® Athlon™, etc.), and other circuitry associated with that. You will also find slots for other peripherals like RAM, ROM, Hard Disk, LAN Card, CPU Fan, Heat Sink, LCD monitor, etc. These peripherals need to be attached to the motherboard separately in order to make the PC/Mac fully functional. Unlike PCs/Mac, single board computers consist of everything on a single board itself! On the board, we have a processor and all other necessary peripherals and circuitry as well. We have onboard RAM, ROM, flash storage, AV ports, Ethernet port, etc. This means that one board is sufficient to act as a full-fledged computer ,even they can boot into an operating system (OS) like Linux, Android, etc. and operate like any other computer. Being lightweight and specific, they have found huge application in smartphones, tablets and other consumer products. These days’ semiconductor manufacturers are building ever powerful processors, which are no less than beasts, thanks to Moore’s Law. These processors, based upon a unique architecture like ARM, Intel x86 or other custom architectures, give whopping performances like 1.2 GHz clock frequency, etc. When combined with 1GB DDR3 RAM, 2GB Flash storage, HDMI/AV port, USB ports, LAN ports, etc. on the same board, it becomes a single board computer! Simply power it up, connect to a display device and boom, You are all set to go… your computer has successfully booted into an OS like Linux, Android, etc. These single board computers are not as powerful as the current day PCs, laptops or Mac, and hence do not dissipate much heat. In addition to that, the processors are designed in order to generate less heat and consume less power. That’s why you can run your smartphone the entire day without charging the battery or cooling it down All the electronic gadgets that you see around – smartphones, tablets, etc. have one such single board computer inside them – their motherboard! Most of them will run Android and iOS (an OS just like Windows, Linux, Mac OSx, etc.). You can download and install apps just like you do on your PC.There are several reasons one might opt to use a single board computer, Portability being one of the major features. You can carry around a small computer like your smartphone in your pocket everywhere you go,These devices are pretty intuitive to use as well. They consume less power and energy as compared to traditional computers. And the most important feature is being cost effective! Being low cost, these products can reach a much larger

4.2 Seriesofsingle-board computers

As an end user (or consumer), examples are all around you – electronic gadgets! Next time you look at any such gadget, Google out its specifications!
As a developer, apart from the gadgets, there are some notable single board computers available in the market for both, hardware and software development. Some of them include Raspberry Pi, The Beagles (BeagleBoard, BeagleBoardxM, BeagleBone, BeagleBone Black), PandaBoard, MK802, MK808, Cubieboard, MarsBoard, Hackberry, Udoo,etc.
Recently, Intel® has also entered into the Open Source world with its Atom™ processor based MinnowBoard.

4.2.1 Raspberry Pi

The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries.Show figure 4.1



The Raspberry Pi is a low cost, single-board computer used to teach computer science.[figure 4.1][01]

Raspberry Pi 3 Model B released in February 2016, is bundled with on board WiFi, Bluetooth and USB boot capabilities.[02] As of January 2017, Raspberry Pi 3 Model B is the newest mainline Raspberry Pi. Raspberry Pi boards are priced between US$5–35. As of 28 February 2017

4.2.2 Hardware

The Raspberry Pi hardware has evolved through several versions that feature variations in memory capacity and peripheral-device support.

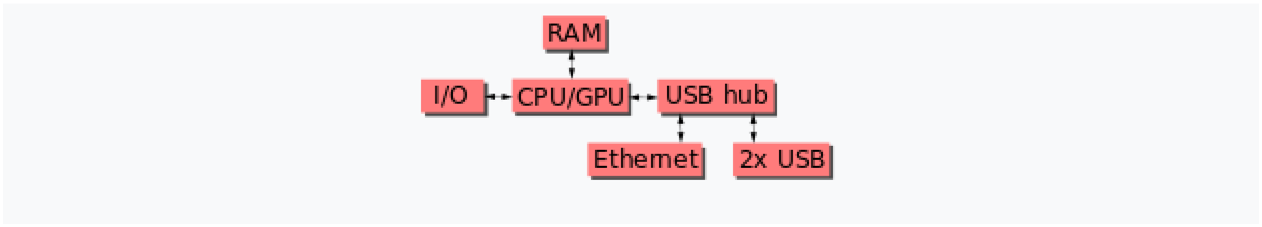


Figure 4.2 [03][adopted]

This block diagram[figure 4.2] depicts Models A, B, A+, and B+. Model A, A+, and the Pi Zero lack the Ethernet and USB hub components. The Ethernet adapter is internally connected to an additional USB port. In Model A, A+, and the Pi Zero, the USB port is connected directly to the system on a chip (SoC). On the Pi 1 Model B+ and later models the USB/Ethernet chip contains a five-point USB hub, of which four ports are available, while the Pi 1 Model B only provides two. On the Pi Zero, the USB port is also connected directly to the SoC, but it uses a micro USB (OTG) port.

4.2.3 Processor

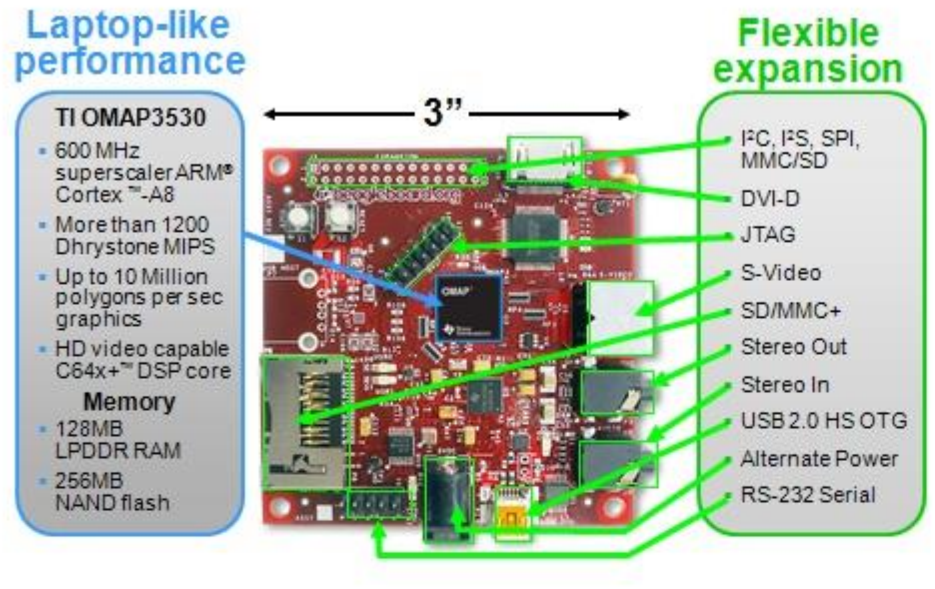
The Broadcom BCM2835 SoC used in the first generation Raspberry Pi is somewhat equivalent to the chip used in first modern generation smartphones (its CPU is an older ARMv6 architecture),[03] which includes a 700 MHz ARM1176JZF-S processor, VideoCore IV graphics processing unit (GPU),[04] and RAM. It has a level 1(L1) cache of 16 KB and a level 2 (L2) cache of 128 KB. The level 2 cache is used primarily by the GPU. The SoC is stacked underneath the RAM chip, so only its edge is visible.The Raspberry Pi 2 uses a Broadcom BCM2836 SoC with a 900 MHz 32-bit quad-core ARM Cortex-A7 processor, with 256 KB shared L2 cache.[05]The Raspberry Pi 3 uses a Broadcom BCM2837 SoC with a 1.2 GHz 64-bit quad-core ARM Cortex-A53 processor, with 512 KB shared L2 cache.[06]

4.2.4 Performance

The Raspberry Pi 3, with a quad-core Cortex-A53 processor, is described as 10 times the performance of a Raspberry Pi 1.[06] This was suggested to be highly dependent upon task threading and instruction set use. Benchmarks showed the Raspberry Pi 3 to be approximately 80% faster than the Raspberry Pi 2 in parallelized tasks. [07]

4.3 Beagle Board

The BeagleBoard is a low-power open-source hardware single-board computer produced by Texas Instruments in association with Digit-Key and Newark element14. The BeagleBoard was also designed with open source software development in mind, and as a way of demonstrating the Texas Instrument's OMAP3530 system-on-a-chip. Show figure [4.3]



BeagleBoarddescribed[figure 4.3][adopted]

4.4 Features

The BeagleBoard measures approximately 75 by 75 mm and has all the functionality of a basic computer.[09] The OMAP3530 includes an ARM Cortex-A8 CPU (which can run Linux, Minix,[10] FreeBSD,[11] OpenBSD,[12] RISC OS,[13] or Symbian; Android is being ported[14]), a TMS320C64x+ DSP for accelerated video and audio decoding, an an Imagination Technologies PowerVR SGX530GPU to provide accelerated 2D and 3D rendering that supports OpenGL ES 2.0. Video out is provided through separate S Video and HDMI connections. A single SD/MMC card slot supporting SDIO, a USB On-The-Go port, an RS-232 serial connection, a JTAG connection, and two stereo 3.5 mm jacks for audio in/out are provided. Built-in storage and memory are provided through a PoP chip that includes 256 MB of NAND flash memory and 256 MB of RAM (128 MB on earlier models).The board uses up to 2 W of power and can be powered from the USB connector, or a separate 5 V power supply. Because of the low power consumption, no additional cooling or heat sinks are required.

References

[01]"Foundation Strategy 2016 - 2018" (PDF). Raspberry Pi.Raspberry Pi Foundation. pp. 3–5. Retrieved 26 November 2016.
[02]"Eben Upton talks Raspberry Pi 3". The MagPi Magazine. 29 February 2016.
[03] "BCM2835 Media Processor; Broadcom". Broadcom.com. 1 September 2011. Archived from the original on 13 May 2012.Retrieved 6 May 2012.
[04] Brose, Moses (30 January 2012). "Broadcom BCM2835 SoC has the most powerful mobile GPU in the world?".Grand MAX.Archived from the original on 18 February 2012.Retrieved 13 April 2012.
[05]"Raspberry Pi 2 on sale now at $35". Raspberry Pi Foundation.Retrieved 5 August 2015.
[06]Upton, Eben (29 February 2016). "Raspberry Pi 3 on sale now at $35 - Raspberry Pi". Raspberry Pi.Retrieved 2016-02-29.
[08] "How Much Power Does Raspberry Pi3B Use? How Fast Is It Compared To Pi2B?". RasPi.TV. RasPi.TV. Retrieved 6 July 2016
[09]September 1, 2008, at the Wayback Machine.
[10]"Minix 3.3.0". Retrieved 2014-09-15.
[11]"creating_bootable_sd_card". Retrieved 2013-05-05.
[12]"OpenBSD/beagle". Retrieved 2013-07-19.
[13]beagleboard.org - RISC OS Details

[14]Porting Android on Beagle Board XM

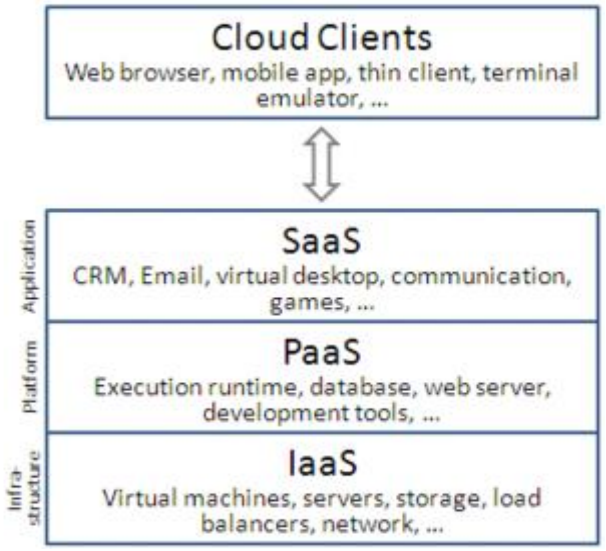
Cloud computing

5 Introduction

Cloud computing is a type of Internet-based computing that provides shared computer processing resources and data to computers and other devices on demand. It is a model for enabling ubiquitous, on-demand access to a shared pool of configurable computing resources (e.g., computer networks, servers, storage, applications and services),[1][2] which can be rapidly provisioned and released with minimal management effort. Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in either privately owned, or third-party data centers[3] that may be located far from the user–ranging in distance from across a city to across the world. Cloud computing relies on sharing of resources to achieve coherence and economy of scale, similar to a utility (like the electricity grid) over an electricity network.

5.1 Service models

Though service-oriented architecture advocates "everything as a service" (with the acronyms EaaS or XaaS or simply aas),[04] cloud-computing providers offer their "services" according to different models, of which the three standard models per NIST are Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).[2] These models offer increasing abstraction; they are thus often portrayed as a layers in a stack: infrastructure-, platform- and software-as-a-service,[05] but these need not be related. For example, one can provide SaaS implemented on physical machines (bare metal), without using underlying PaaS or IaaS layers, and conversely one can run a program on IaaS and access it directly, without wrapping it as SaaS.


Cloud Clients
Web browser, mobile app, thin client, terminal emulator, ...
SaaS
CRM, Email, virtual desktop, communication, games, ...
PaaS
Execution runtime, database, web server, development tools, ...
IaaS
Virtual machines, servers, storage, load balancers, network, ...
Application
Platform
Infra-structure

Cloud computing service models arranged as layers in a stack[figure 5.1][adopted]

The NIST's definition of cloud computing defines the service models as follows:[2]

- Software as a Service (SaaS). The capability provided to the consumer is to use the provider’s applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.
- Platform as a Service (PaaS). The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.
- Infrastructure as a Service (IaaS). The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls).

5.2 Cloud clients

Users access cloud computing using networked client devices, such as desktop computers, laptops, tablets and smartphones and any Ethernet enabled device such as Home Automation Gadgets. Some of these devices—cloud clients—rely on cloud computing for all or a majority of their applications so as to be essentially useless without it. Examples are thin clients and the browser-based Chromebook. Many cloud applications do not require specific software on the client and instead use a web browser to interact with the cloud application. With Ajax and HTML5 these Web user interfaces can achieve a similar, or even better, look and feel to native applications. Some cloud applications, however, support specific client software dedicated to these applications (e.g., virtual desktop clients and most email clients). Some legacy applications (line of business applications that until now have been prevalent in thin client computing) are delivered via a screen-sharing technology.

NOTE:-
Will using Google cloud API

INTERNET OF THINGS

6.1 Introduction:

The Internet has initially started as the "Internet of Computers", a global network enabling services that now include the World Wide Web (WWW), File Transfer Protocol (FTP) and others allowing computers and hence users to communicate with each other and exchange information. In the recent years, the device processing power and storage capacity are increasing while at the same time technology is making devices pervasive, mobile and wearable. In addition, networking technologies are evolving and communication electronic systems are becoming smaller and cheaper. Devices are increasingly fitted with sensors and actuators, creating environments where the former are connected to various networks. Devices can sense, compute, act and thus intelligently become parts of the so-called 'Internet of Things'

6.2 INTERNET OF THINGS definition

There are several definitions for the Internet of Things (IoT) that also explain what are the main functionalities of it and what we should expect from when connecting 'Things' with each other and with the Internet. Some people suggest, that the "Internet of Things can be seen as a potentially integrated part of the 'Future Internet'. Wikipedia defines IoT as: "A part of a dynamic global network infrastructure with self-configuring capabilities based on open and interoperable communication protocols where physical and virtual 'Things' interact with each other. These 'Things' have specific identities, physical attributes, virtual 'personalities' and use intelligent interfaces. They are able to interact and communicate among themselves and with the environment by exchanging data and information 'sensed' about the environment, while reacting autonomously to the 'real/physical world' events and influencing them by running processes that trigger actions and create services with or without our direct intervention. Interfaces in the form of services facilitate interactions with these 'smart things' over the Internet, query and change their state and any information associated with them."

6.3 The Basic Concepts

When we are referring to 'Things', we talk about devices and everyday objects, from small   ones (like wrist watches and medical sensors) to really big ones (like robots, cars and buildings).   All such contain devices that interact with users by generating and retrieving information about   and from their environment (see Figure 1-1). They also contain hardware that allows them to control outputs (like relay switches or digital ports).

No matter what definition of the Internet of Things you may find the main concept behind every IoT technology and implementation is the same: devices are integrated with the virtual world of the Internet and interact with it by tracking, sensing, and monitoring objects and their environment. Users and developers add components, give them sensing and networking capabilities, program them to perform the aforementioned tasks and build Web applications that interacting with the devices.

The features of a device that can act as a member of an IoT network can be summarized into the following:

- Collect and transmit data: The device can sense the environment (e.g., your home or your body) and collect information related to it (e.g., temperature and lighting conditions) and transmit it to a different device (can be your mobile phone or your laptop) or to the Internet.
- Actuate devices based on triggers: It can be programmed to actuate other devices (e.g., turn on the lights or turn off the heating) based on conditions set by you. For instance, you can program the device to turn on the lights when it gets dark in your room.
- Receive information: One unique characteristic for IoT devices is that they can also receive information from the network they belong to (i.e. other devices) or through the Internet (e.g., information from you like new triggers, new status of operation and in some cases new functionality).
- Communication assistance: IoT devices that are members of a device network can also assist in communication (i.e. data forwarding) between other nodes of the same network. Think of them as messengers for devices (nodes) that are not very close to an endpoint (e.g., your router) in order to get direct information from

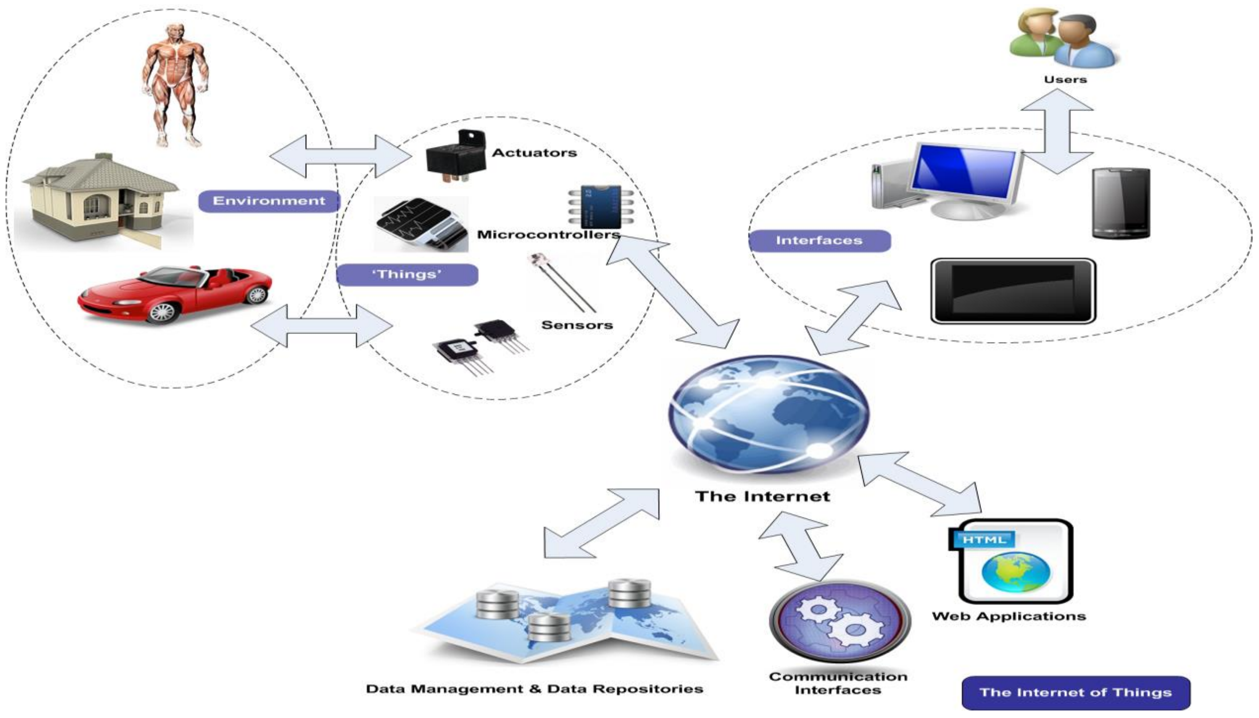


Figure 6.1.An illustration of the 'Internet of Things'. 'Things' consisting of various sensors andactuators interact with environment and the Internet allowing users to manage them and their data over various interfaces.

6.4 Interaction with the internet

What makes iot device different than ordinary sensor devices is basically the ability to communicate (most usually) directly or indirectly to the Internet. So what are the main reasons a device would need to communicate with an Internet service? What kind of service would that be and what kind of features would it have? Firstly, sensors generate a lot of data that needs somehow to be managed. Usually, embedded memory is quite limited so people utilize alternative solutions like storing data on memory cards, or in computers in cases sensors are directly connected to them. Since sensors can be integrated to devices with further networking capabilities, why not to store the information online? Through this way we can solve the limited storage issue and at the same time we can access the data anywhere, anytime using appropriate web applications.

In addition, most of these web applications provide interfaces for data exchange between
Other applications and mostusefully,with mobile applications. So it is possible that your iphone
Or Android mobile can talk directly to your IoT device!
We can also use the web platforms to send data back to the devices. The data can be triggers,
Instructions for activating actuators and switches, or simply information from the Internet, like a
weather channel or a Twitter account that can displayed through an interface like an LCD screen.
The web-based platforms that enable the aforementioned functionality for data management
And information exchange are based on Cloud computing.
In order for things to talk to each other and to the Internet, they need to have specific
Abilities and serve more specific functions. The following section discuses what are the
Components that are included in IoT devices that enable them to sense the environment, act and
Communicate.

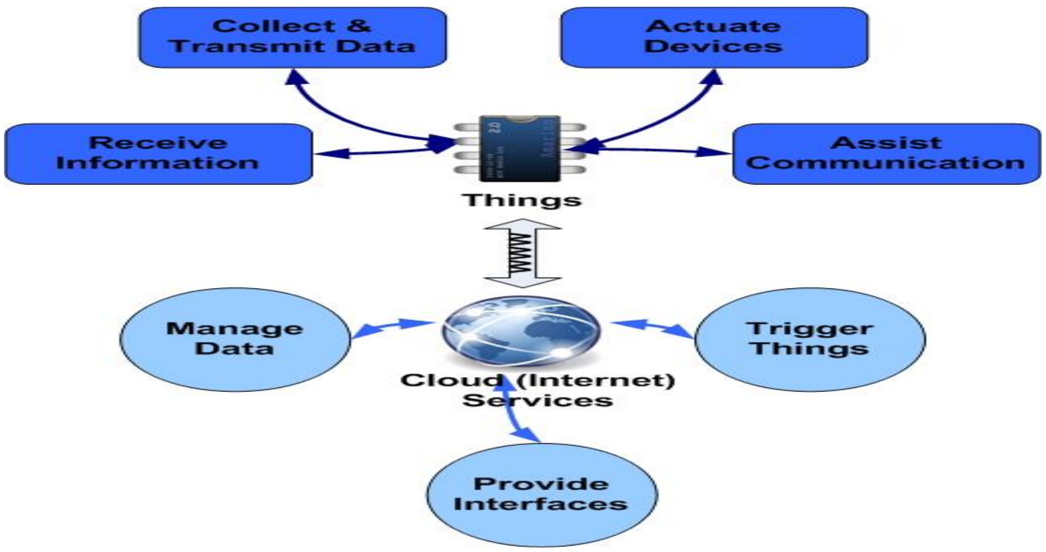


Figure 6.2.Illustration of the main features of the 'Things' in IoT networks and the respective Internet services.

6.5 Major Components of IoTDevices

6.5.1 Control Units

Usually in the world of IoT, devices utilize a microcontroller as the main control unit responsible for the aforementioned operations.
A microcontroller can be considered as a small computer on a single integrated circuit (often abbreviated as IC) containing a processor core, memory, and 5 programmable input and output peripherals .

Other important parts of the microcontrollers are the analog to digital converters (ADC) that are used to convert incoming analog (signal) data into a form that the processor can recognize. In addition to the converters, many such control units include a variety of timers as well.

6.5.2 Sensors

Sensors are devices that can measure a physical quantity (like temperature, humidity, etc.) and convert it into a signal, which can be read and interpreted by the microcontroller unit. They are the devices that are most likely attached to the input pins of the microcontroller in our embedded Projects. Generally, most sensors fall into two categories; analog and digital sensors.

6.5.2.1 An analog sensor

Such as a thermostat (which actually is a resistor changing its resistance Based on temperature used in digital thermometers), or a humidity sensor, is connected into Circuits so that it generates a specific voltage range, usually between 0 volts to 5 volts. This Output goes to the analog input pin of the microcontroller unit (see Figure 1-4). The latter uses the appropriate analog-to-digital (A/D) circuit to convert voltage into a numeric value that we can later read process and send to the Internet. Figure 1-4 shows how an analog sensor can be connected to a microcontroller device. All analog sensors usually come with 3 connection pins, one for receiving voltage (Vic), one for ground connection (GND) and the output voltage pin (OUT).

6.5.2.2 Digital sensors

Generate what is called a Discrete Signal. Such a signal has different (usually Binary only) states, like the logic gate states 0 and 1, which are represented by high and low Current.
For example, consider a push button switch (which by the way is one of the simplest Forms of digital sensors): it has two discrete values. It is on, or it is off.
Some more complicated Sensors come also with a digital interface. This means that instead of generating a signal representing the voltage change between 0-5 based on their measurement, they actually generate a stream of bits (0 and 1 states).

6.5.3 Communication Modules

The communication modules are parts of the device, which are responsible for the Communication with the rest of the Iota
They provide connectivity according to the Wireless or wired communication protocol they are designed for (such protocols are presented in the rest of this chapter).
They usually consist of embedded electronic modules that implement the communication (i.e. transform the information received in bits and bytes to radio waves or Signals that are transferred by wire respectively). Wireless modules usually consist also of an Antenna for achieving maximum range, either external or internal for optimizing the size of the Device.

Most likely the communication between Iota devices and the Internet is performed in two Ways:

- There is an internet-enabled intermediate node acting as a gateway, the device is commonly connected to the Computer and sends data to it using e.g., a USB port. The computer receives the data and using appropriate software it forwards it to the Internet. In the second case, things are much simpler devices can function and communicate more autonomously.

- The Iota device has direct communication to the Internet; communication can be made according to the wireless technology used. Most Modules that support wireless (like Wife, Bluetooth and Sigsbee) and wired technologies (like the Ethernet) also support the TCP/IP protocol. The TCP/IP protocol (for those unfamiliar with it) is the way computers, mobile phones and internet-enabled devices communicate with each other and to the Internet. It defines all communication essentials (e.g., how devices are identified by each other, using IP addresses; how information is split into small packets and transmitted; etc.). It also takes care of all connection setup issues and ensures information is properly delivered by Retransmitting data whenever this is considered necessary.

What about inter-device communication?

The communication between the mainUnits and the various communication modules is performed in most cases (and in the case of our Projects) using the serial protocol. Serial is a device communication protocol that is standard on PC and electronic device. The concept of serial protocol is simple. Each device has Two ports (or connectors), one for transmitting data, usually annotated as Tax, and one for receiving data, usually annotated as Rx. Also a ground and power connection are required and are provided by the power source. Each serial port sends and receives bytes of information one bit at a time. So, in order to communicate, both control units (i.e. microcontrollers) and Communication modules share a pair of Rx/Ports, although this is slower than parallel communication, which allows the transmission of an Entire byte at once, it is simpler and you can use it over longer distances. Because serial is Asynchronous, the port can transmit data on one line while receiving data on another. A brief presentation of communication enabling technologies for devices follows in the "Communication Technologies" section.

6.5.4 Power Sources

Every electronic device needs electric power to properly function. The electric power is the result of the potential difference between two points, otherwise known as voltage, and the electric Flowing through a circuit. In most electronic devices and also in the projects we are dealing with in this book, the two points are referred as the voltage input point (usually notated as VIN or Vic) and the ground (usually notated as GND).In small devices the current is usually produced by sources like batteries, thermocouples and solar cells. Batteries are electrochemical components that convert chemical power to electrical generating direct current (DC).Wearable and mobile devices are mostly powered by small lightweight batteries that can also be recharged (like in Figure 6-3) for longer life duration. By utilizing simple circuits and appropriate programming of microcontrollers, devices can be aware of their battery status and alert users when they need to be recharged or replaced.

Figure 6.3.Probably the smallest rechargeable consumer battery. Weighs only 330mg and operates at 3.6V (image courtesy of Power Stream)

When selecting the appropriate battery source for your own projects, you look for the battery's operational voltage and capacity. The latter is associated with the total energy stored within the battery. Capacity in general is measured in watt per hours (Who), kilowatt per hours (kWh) or ampere per hours (Air). For sensors and other components of IoT projects you will build, the most common measure of battery capacity is Ah.To give you an example about capacity and its actual meaning, if your IoT device needs 50mA to operate and your battery provides 50mAH, the latter will power your device adequately for about 1 hour before needing recharging.

6.6 Communication Technologies

'Things' need to talk to each other and also talk to the Internet in order to exchange sensor outputs, triggers, status messages etc. In order to do so, devices need to integrate a wireless (preferably) or a wired communication system. The major communication technologies that can be utilized by such devices are summarized in this section. In addition to the brief description of the technologies, samples of electronic modules that enable the respective communication are also presented. The modules selected are characterized by the special interfaces they have for connecting directly to microcontrollers and platforms like the 'arduino' open-source electronics prototyping platform.

6.6.1 RFID

The Radio Frequency Identification (RFID) technology has been initially introduced for identifying and tracking objects with the help of small electronic chips, called tags. It is the most common technology behind asset tracking (that tells you where your mail parcel is before arriving its destination) and identifying objects (e.g., in automatic toll collection).RFID tags are categorized into passive, active and battery assisted passive (BAP). The passive tags do not have a power source (battery) and thus cannot transmit and information on their own. They are powered-activated by the RFID reader and transmit only a small amount of information (usually an identification number (ID) of the tag). Active tags on the contrary have their own battery and can broadcast data continuously. A BAP tag can be considered as a hybrid: it carries a battery but only transmits information in presence of an RFID reader. The battery helps them to transmit their signal in longer distance than the passive tags (restricted to a few cm). Most RFID tags consist of an integrated circuit for storing and processing the information, the essential radio frequency (RF) components for transmitting the information wirelessly and an antenna.RFID has been initially characterized as the enabling communication power for the Internet of Things, due to its low cost, high mobility and efficiency in identifying devices and objects. Despite RFID being very popular for device identification and some information exchange (e.g., RFID-based temperature sensors) it cannot alone support the creation of IoT networks since it cannot provide any direct or indirect (e.g., through a gateway) communication to the Internet. The device proximity is also another drawback.

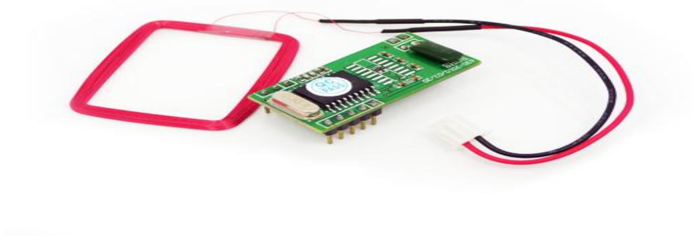


Figure 6.4 RFID module and its antenna. It can directly communicate with the rpi. using the Serial protocol. (image courtesy of Seeedstudio).

RFID has been initially characterized as the enabling communication power for the Internet of Things, due to its low cost, high mobility and efficiency in identifying devices and objects. Despite RFID being very popular for device identification and some information exchange (e.g., RFID-based temperature sensors) it cannot alone support the creation of IoT networks since it cannot provide any direct or indirect (e.g., through a gateway) communication to the Internet. The device proximity is also another drawback.

6.6.2 Bluetooth

You are using the Bluetooth technology to connect your wireless headset with your mobile phone, or to transfer photos from the latter to your computer. Bluetooth is a technology standard for exchanging data over short distances (using short wavelength radio transmissions in the ISM band from 2400-2480 MHz) from fixed and mobile devices, creating personal area networks. Bluetooth has been one of the first wireless communication protocols designed with lower power consumption for replacing short-range wired communications (in computer peripherals, mobile phone accessories, etc.). One important feature of Bluetooth is that devices can discover and communicate with each other without the need to be in visual line of sight (like in infrared communication), which is very important when using Bluetooth as a network technology for sensor systems deployment.Bluetooth is named after the 10th century Danish King Herald Blatant, which translates as Harold Bluetooth in English!
It is commonly used for connecting small devices with each other, due to the fact that it can support automatically the creation of peer networks (i.e. networks of devices that exchange and forward information) and provides communication functionality with low power consumption. The latter is very important for the case of IoT since many of the devices that one would like to interconnect to the IoT (sensors, actuators, etc.) have limited power resources. However on major drawback of Bluetooth is that it cannot provide direct connectivity to the Internet. One has to provide an intermediate node, e.g., a PC that will act as a gateway to the outer world.

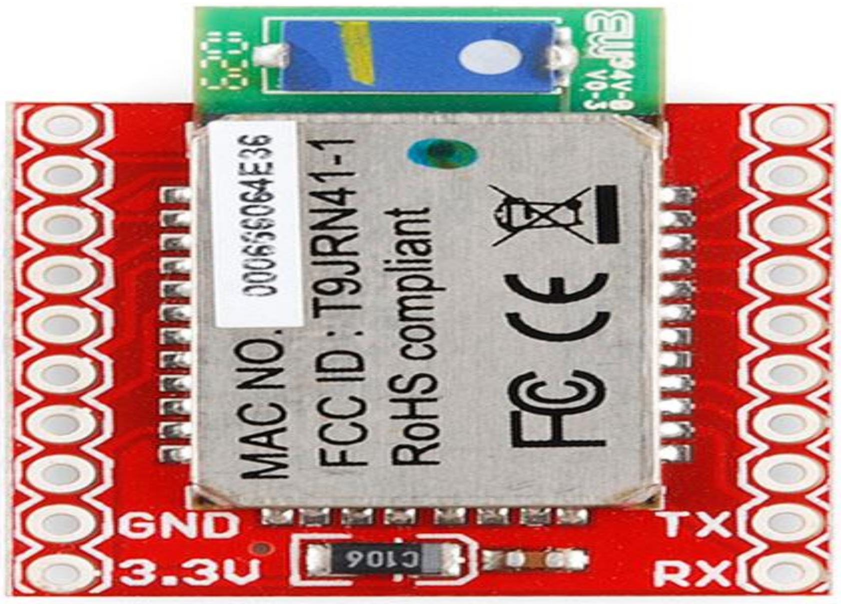


Figure 6.5 Bluetooth communication module

6.6.3 Zigbee

Zigbeeis one of the latest and most advanced wireless technologies being widely integrated into home automation & smart devices worldwide. It has been specifically developed as an open global standard to address the unique needs of low-cost, low-power wireless networks for communication between devices (also known as machine-to-machine or M2M networks).
The Zigbee standard operates in unlicensed bands including 2.4 GHz, 900 MHz and 868 MHza Sigsbee module can have a low consumption of 50mA. For instance, a rechargeable battery of 850mAH can provide about 17 hours of continuous operation for such module. Maximum data rate is about 250kbps and communication range can vary from 100m to 1km (maximum) depending on the output power (in small projects we usually use 1mW modules that provide maximum of 100m range).Compared to Bluetooth, Sigsbee provides better power efficiency, and higher range, making it
Thus a better wireless technology to consider for your IoT network. It also allows the automatic creation of peer networks and as a matter of fact it is consider being more extensible in this context.
However it still requires a gateway with Internet connectivity (e.g., a laptop) that will forward information from the Internet to the Zigbee network and vice versa.

Figure 1-9. ZigBee communication module

6.6.4 Wifi

Wifi, also known as the IEEE 802.11x standard, is the most common way to connect devices wirelessly to the Internet. Your laptop, Smartphone and Tablet PC are equipped with Wife interfaces and talk to your wireless router and provide you this way access to the Internet.
The commercially available Wi-Fi modules can be directly integrated to an IoT device and provide instant connectivity. The major advantage over the other wireless technologies is the fact that Wi-Fi networks are very easy to establish and thus IoTdevices with Wi-Fi modules can have direct connection to the Internet. One drawback is the fact that this technology (which was at no means designed for IoT networks) is more power demanding than the others.

Figure 6.6 WiFi communication module

6.6.5 RF Links

Another option to connect devices and make them talk is utilize simple radio frequency (RF) interfaces. The latter are quite cheap and small (ideal when size matters) and can provide communication range between 100m and 1km (depending on the transmission power and the antenna used).
RF communication modules are connected to microcontroller and devices via through serial ports as the rest of the modules we have examined. However they do not provide any implementation of the TCP/IP communication protocol (or any other protocol).
This means that if you want your devices to communicate you have to create your own protocol for establishing communication, identifying devices with each other and make sure all the information you have transmitted is delivered. Data rates are quite low (up to 1Mpbs) and you also need an Internet-enabled gateway that will provide access to your devices for making a completeIoT network.
An RF transceiver module (it can be used both as a receiver and transmitter).
It has low power consumption, can support data rates up to 1Mbps and can be easily interfaced to your arduino

Figure 6.7 An RF transceiver module

6.6.6 Cellular Networks: The Mobile Internet

The ‘Mobile Internet' refers usually to access to the Internet from a mobile device, such as a Smartphone or laptop through a mobile broadband network. The mobile broadband network is based on cellular communication, same technology that is used in our mobile phones for serving our calls and text messages. It can provide direct Internet connectivity to a variety of data rates.
Various network standards have existed for serving mobile Internet. GPRS, 3G, IMAX, and LTE (one of the 4G technologies) are a few to name. Depending on the standard and the available network coverage, connection speeds can go from 80Kbps (GPRS) to a few Mbps (3G and 4G).
Due to the complexity of the communication protocol and the information coding, in addition to high power requirements in cases where reception signal is low, the battery consumption of mobile Internet – enabled devices is an issue. Think of how fast the battery of your cell phone is drained when you browse the Internet. Still it is a good option for connecting devices directly to the Internet, since small GPRS modules for the Arduino are available and connectivity does not require further infrastructure (e.g., Internet connected laptop like in case of ZigBee or Bluetooth).

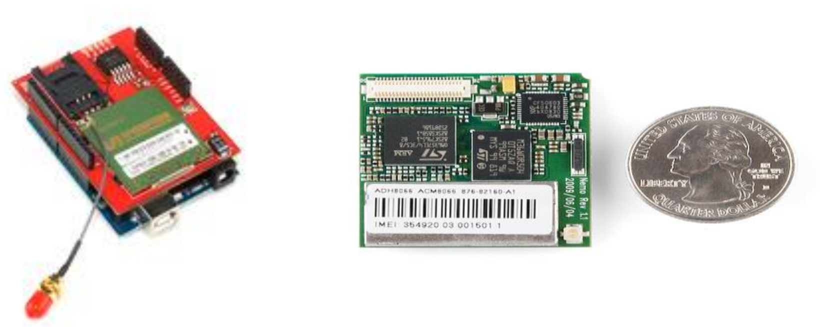


Figure 6.8. GPRS shield for your Arduino on the left. The GPRS module on the right

6.6.7 Wired Communication

Devices can definitely talk to each other and to the Internet using wired infrastructures, given the appropriate network interface. What is the best way? Since the very beginning of the Internet era to nowadays, your desktop computer has at least one Ethernet port.
The Ethernet protocol is very well established in computer communications. It does not require as much power as the wireless communications do, it can achieve very high data rates and most importantly it is so common in computer communications that all you need to do is plug an Ethernet cable to its device and get online. No worries about signal availability. It does not provide any mobility, but its advantages were the main reason it was one of the first networking technologies adopted in microcontroller platforms like the Arduino.
An Ethernet module. Can be directly connected to any sensor device and provide wired network functionality. On the right: An Arduino with embedded Ethernet interface

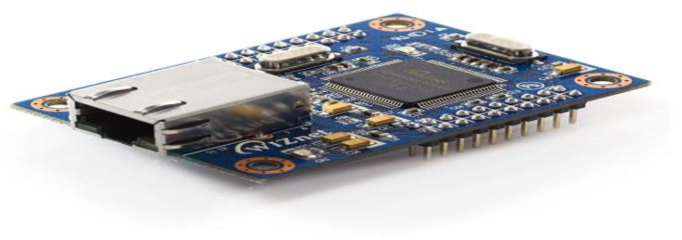


Figure 6.9. An Ethernet module

References

[01]Hassan, Qusay (2011). "Demystifying Cloud Computing" (PDF). The Journal of Defense Software Engineering.CrossTalk. 2011 (Jan/Feb): 16–21. Retrieved 11 December 2014
[02]Peter Mell and Timothy Grance (September 2011). The NIST Definition of Cloud Computing (Technical report). National Institute of Standards and Technology: U.S. Department of Commerce. doi:10.6028/NIST.SP.800-145. Special publication 800-145.
[03] M. Haghighat, S. Zonouz, & M. Abdel-Mottaleb (2015).CloudID: Trustworthy Cloud-based and Cross-Enterprise Biometric Identification. Expert Systems with Applications, 42(21), 7905–7916.
[04] Kurdi, Heba; Li, Maozhen; Al-Raweshidy, H. S. (2010). "Taxonomy of Grid Systems".In Antonopoulos, Nick. Handbook of Research on P2P and Grid Systems for Service-Oriented Computing: Models, Methodologies and Applications. IGI Global research collection.IGI Global.p. 34.ISBN 978-1-61520-687-2.Retrieved 2015-07-29. Nowadays Service-Oriented Architecture (SOA) has become as [sic] the main architectural model of many IT initiatives including grid, cloud and everything as a service (Essa\XaaS\aas) computing.
[05]AlcarazCalero, Jose M.; König, Benjamin; Kirschnick, Johannes (2012). "Cross-Layer Monitoring in Cloud Computing". In Rashvand, Habib F.; Kavian, Yousef S. Using Cross-Layer Techniques for Communication Systems.Premier reference source.IGI Global.p. 329.ISBN 978-1-4666-0961-7.Retrieved 2015-07-29. Cloud Computing provides services on a stack composed of three service layers (Hurwitz, Bloor, Kaufman, &Halper, 2009): Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).

Glossary

API - Application Programming Interface
aas - as a service
IaaS - Infrastructure as a Service
PaaS -Platform as a Service
SaaS -Software as a Service
NIST - National Institute of Standards and Technology

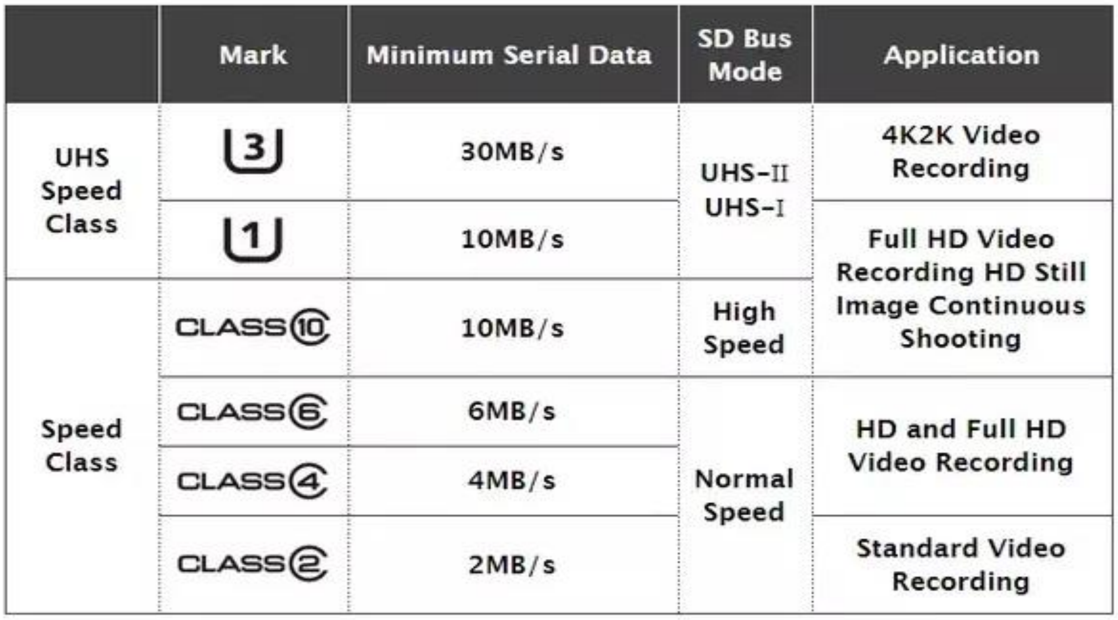
CHAPTER THREE

PROTOTYPE AND SYSTEM IMPLEMENT

1 Raspberry pi

observe should take when using RPI

- NOT work in an area where the floor is carpeted.
- a moment before touching the RPi, touch a substantial size metal object first (ex. a computer case or metal cabinet).
- when handling the RPi, hold the board by the edges and only let your fingers touch the board edge or the outside of the large connectors on the board, do not let your fingers touch chips or pins.
- Choose memory card in class 10 that’s more speed in work from other memory .
  Show Figure 1.1
- shutdown the RPi (sudo shutdown -h now) and disconnect power before boarding a circuit.



	Mark	Minimum Serial Data	SD Bus Mode	Application
UHS Speed Class	U3	30MB/s	UHS-II UHS-I	4K2K Video Recording
	U1	10MB/s		Full HD Video Recording HD Still Image Continuous Shooting
Speed Class	CLASS 10	10MB/s	High Speed	
	CLASS 6	6MB/s	Normal Speed	HD and Full HD Video Recording
	CLASS 4	4MB/s		
	CLASS 2	2MB/s		Standard Video Recording

Figure 1.1 different between memory card [adopted]

6.7 Operating system

Raspbian[figure1.2] is the Foundation's official supported operating systems will use it as a os .

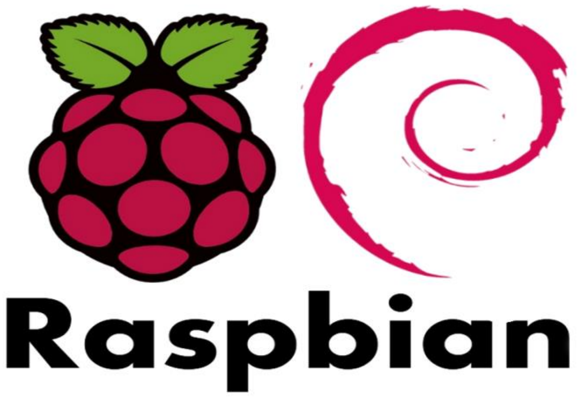


Figure 1.2[adopted][01]

6.7.1 Installation

- download latest .zip version of Raspbian from www.raspberrypi.org, ex: "2017-09-24-raspbian-jessie.zip", then unzip
- download and install Win32DiskImager

(either follow the link from this page on the Raspberry Pi official site "https://www.raspberrypi.org/documentation/installation/installing-images/windows.md" or Google "Win32 Disk Imager SourceForge"), during installation check the "Create a desktop icon", for all other choices the defaults are ok.

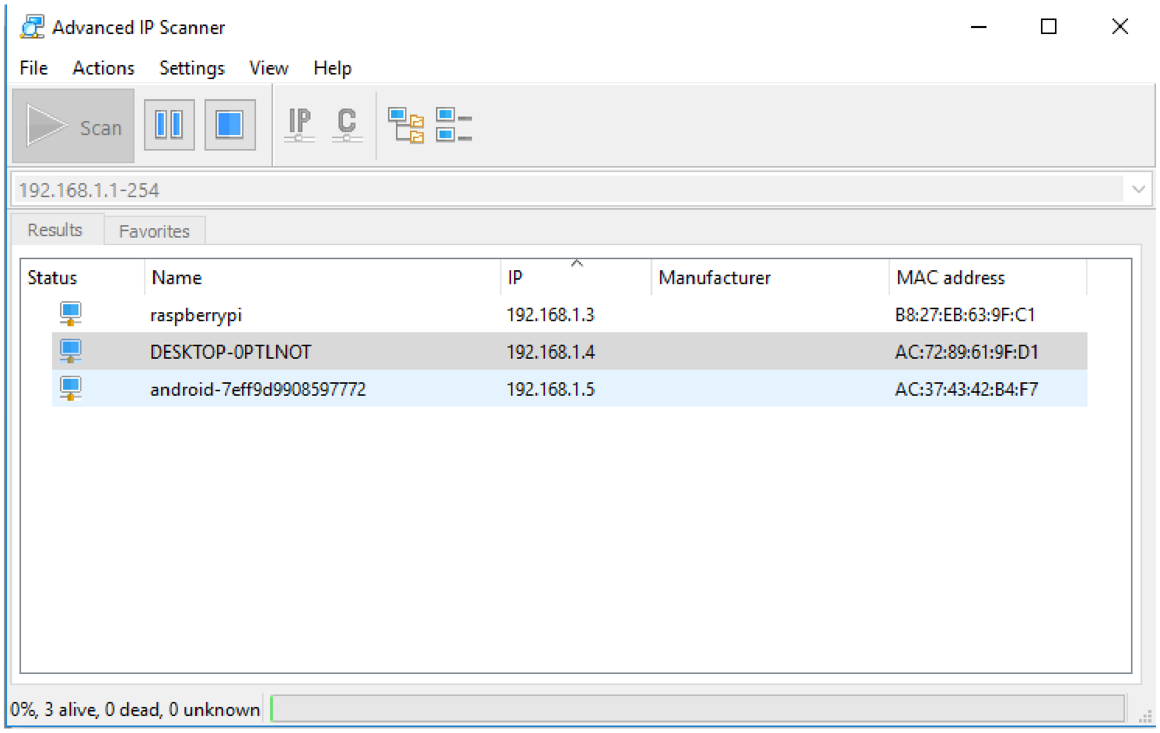
- insert and format your SD card (default settings are ok for the format).
- open Win32DiskImager and flash Raspbian to the SD card.

6.7.2 Software require to connect with rpi from pc

- Putty to connect with pi through *SSH protocol .*
- Advanced IP Scanner to find the ip address of pi .
- Remote Desktop Connection to connect with by in GUI mode.

6.7.3 First Time Boot-up

1- Open Advanced IP Scanner in your pc .show figure 1.3 Then enter scan to find the ip address of rpi



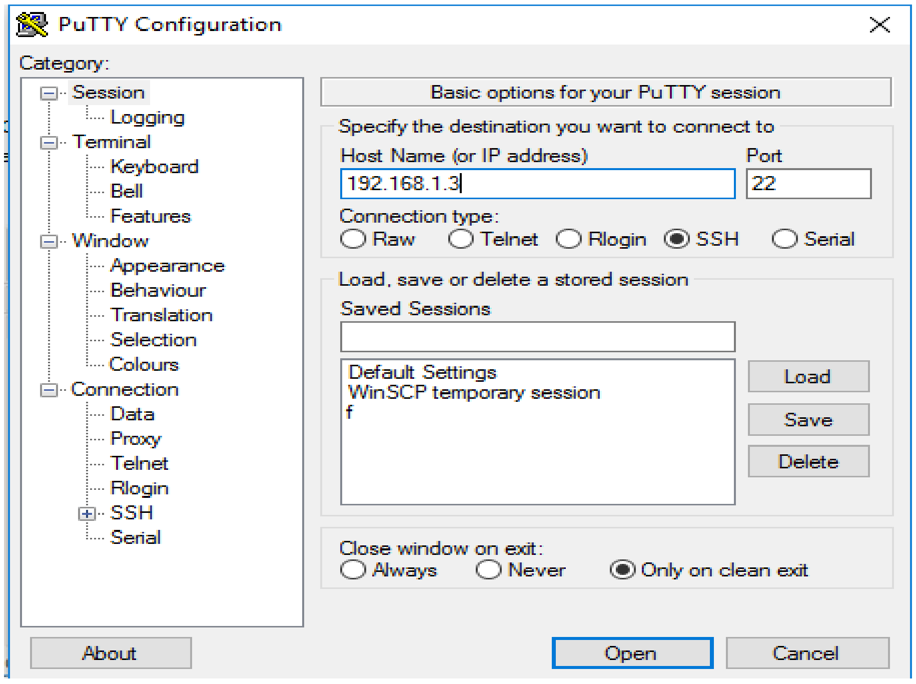
192.168.1.1-254

Status	Name	IP	Manufacturer	MAC address
	raspberrypi	192.168.1.3		B8:27:EB:63:9F:C1
	DESKTOP-0PTLNOT	192.168.1.4		AC:72:89:61:9F:D1
	android-7eff9d9908597772	192.168.1.5		AC:37:43:42:B4:F7

0%, 3 alive, 0 dead, 0 unknown

Figure 1.3 scan theip of raspberry pi

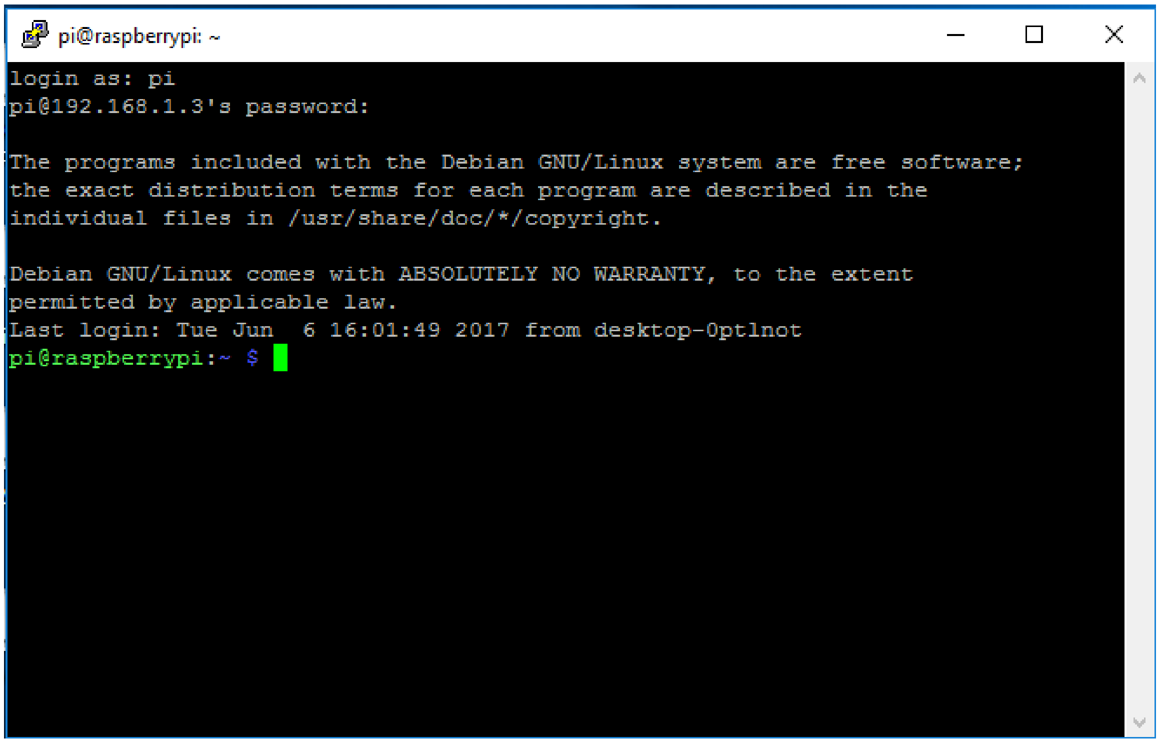
2- Open putty and choose ssh protocol then enter open .show figure 1.4



Host Name (or IP address): 192.168.1.3
Port: 22
Connection type: Raw Telnet Rlogin [✓ SSH] Serial
Default Settings
WinSCP temporary session
f
Close window on exit: Always Never [✓ Only on clean exit]

Figure 1.4 putty

3- Then enter default user name and password (pi & raspberry) and enter to be connect with pi in console mode.show*figure 1.5*



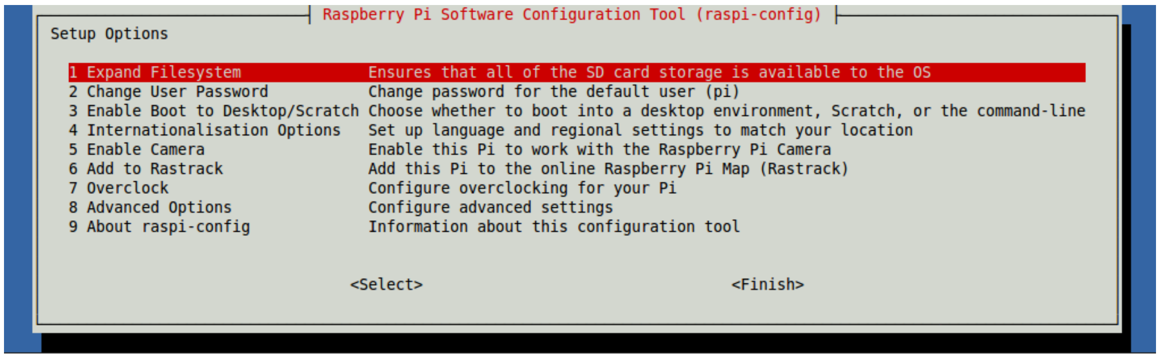
```
login as: pi
pi@192.168.1.3's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jun  6 16:01:49 2017 from desktop-0ptlnot
pi@raspberrypi:~ $
```

4- enter in super user through write

```
sudosu–
```

5- Expand Filesystem figure 1.6 then update and upgrade system

```
sudoraspi-config
sudo apt-get update
sudo apt-get upgrade
```



```
Raspberry Pi Software Configuration Tool (raspi-config)
Setup Options

1 Expand Filesystem             Ensures that all of the SD card storage is available to the OS
2 Change User Password          Change password for the default user (pi)
3 Enable Boot to Desktop/Scratch Choose whether to boot into a desktop environment, Scratch, or the command-line
4 Internationalisation Options  Set up language and regional settings to match your location
5 Enable Camera                 Enable this Pi to work with the Raspberry Pi Camera
6 Add to Rastrack               Add this Pi to the online Raspberry Pi Map (Rastrack)
7 Overclock                     Configure overclocking for your Pi
8 Advanced Options              Configure advanced settings
9 About raspi-config            Information about this configuration tool

                 <Select>                        <Finish>
```

Figure 1.6 software configuration tool

6- Install the xrdp server to connect pi with remote desktop

```
apt-get install xrdp
```

7-open remote desktop in your pc and enter the ip address of pi then the user name and address figure 1.7

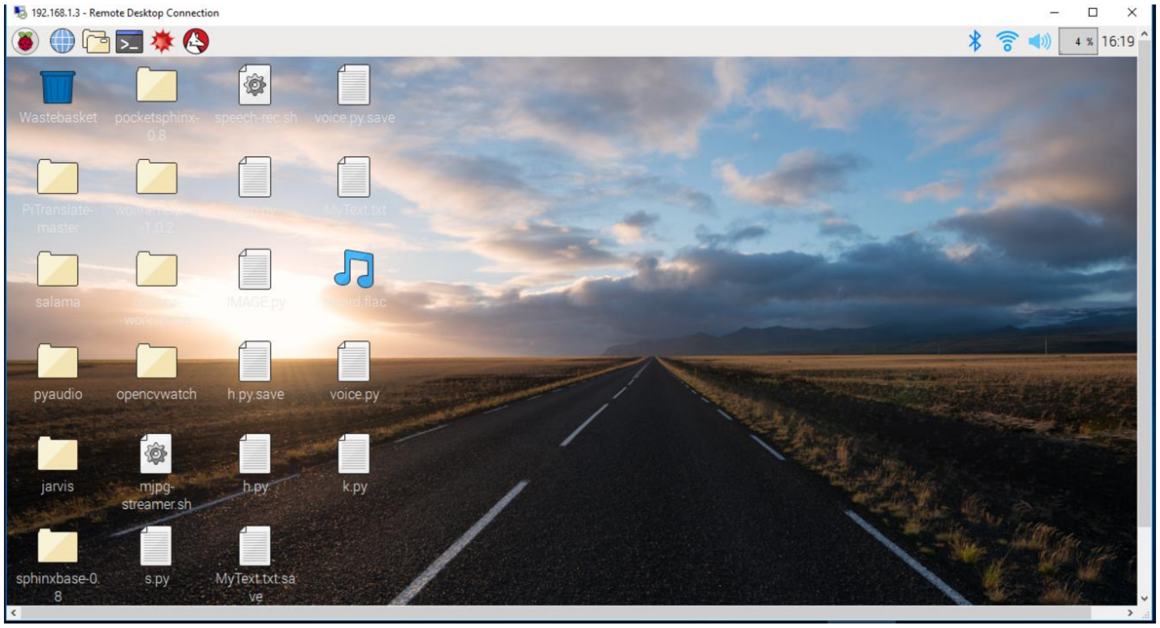


Figure 1.7 the desktop of rpi through remote desktop

Note :-
1-the rpi and pc must be in same network.
2-in every time need to connect with by make steps 1 and choose the mode to connect in GUI (remote desktop)step 7 or console (putty) step 2 only not make the all steps

7 Opencv[03]

OpenCV (Open Source Computer Vision Library) is released under a BSD license and hence it's free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform.

Adopted all around the world, OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 14 million. Usage ranges from interactive art, to mines inspection, stitching maps on the web or through advanced robotics.

7.1 Installation in Linux

These steps have been tested for Ubuntu 10.04 but should work with other distros as well.

7.2 Required Packages

- GCC 4.4.x or later
- CMake 2.6 or higher
- Git
- GTK+2.x or higher, including headers (libgtk2.0-dev)
- pkg-config
- Python 2.6 or later and Numpy 1.5 or later with developer packages (python-dev, python-numpy)
- ffmpeg or libav development packages: libavcodec-dev, libavformat-dev, libswscale-dev
- [optional] libtbb2 libtbb-dev
- [optional] libdc1394 2.x
- [optional] libjpeg-dev, libpng-dev, libtiff-dev, libjasper-dev, libdc1394-22-dev

The packages can be installed using a terminal and the following commands or by using Synaptic Manager:

```
[compiler] sudo apt-get install build-essential

[required] sudo apt-get install cmakegit libgtk2.0-dev pkg-
configlibavcodec-devlibavformat-devlibswscale-dev

[optional] sudo apt-get install python-dev python-numpy libtbb2 libtbb-
devlibjpeg-devlibpng-devlibtiff-devlibjasper-dev libdc1394-22-dev
```

7.3 Getting OpenCV Source Code

You can use the latest stable OpenCV version available in *sourceforge* .

7.3.1 Getting the Latest Stable OpenCVVersion

- Go to our “https://sourceforge.net/projects/opencvlibrary/”
- Download the source tarball and unpack it.

7.3.2 Getting the Cutting-edge OpenCVfrom the Git Repository

Launch Git client and clone “https://github.com/opencv/opencv”

In Linux it can be achieved with the following command in Terminal:

```
cd ~/<my_working _directory>

git clone https://github.com/opencv/opencv.git
```

7.4 Building OpenCV from Source Using CMake, Using the Command Line

1- Create a temporary directory, which we denote as <cmake_binary_dir>, where you want to put the generated Makefiles, project files as well the object files and output binaries.
2- Enter the <cmake_binary_dir> and type

```
cmake [<some optional parameters>] <path to the OpenCV source
directory>
```

For example

```
cd ~/opencv
mkdir release
cd release
cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local
..
```

3- Enter the created temporary directory (<cmake_binary_dir>) and proceed with:

```
make
sudo make install
```

Note:-

Use cmake -DCMAKE_BUILD_TYPE=RELEASE -DCMAKE_INSTALL_PREFIX=/usr/local .. , without spaces after -D if step 2 do not work.

If the size of the created library is a critical issue (like in case of an Android build) you can use the install/strip command to get the smallest size as possible. The *stripped* version appears to be twice as small. However, we do not recommend using this unless those extra megabytes do really matter.

4- opencv.conf will be blank, add the following line, then save and exit nano

```
sudonano /etc/ld.so.conf.d/opencv.conf
/usr/local/lib        # enter this in opencv.conf, NOT at the command line
(leave a blank line at the end of opencv.conf)
```

5- add the following lines at the bottom of bash.bashrc

```
sudoldconfig
sudonano /etc/bash.bashrc
>>PKG_CONFIG_PATH=$PKG_CONFIG_PATH:/usr/local/lib/pkgconfig    , enter these at the bottom of bash.bashrc, NOT at the command line
```

6- Restart the rpi

```
sudo shutdown -r now
```

7- verify the OpenCV install (in python shell) with last version

```
>>>import cv2
>>> cv2.__version__
```

7.5 Color Filtering with opencv[04]

we're going to cover how to create a sort of filter, revisiting the bitwise operations, where we will filter for specifically a certain color, attempting to just show it. Alternatively, you could also specifically filter out a specific color, and then replace it with a scene, like we did with replacing a ROI (region of image) with something else, much like how a green screen works.

In order to filter like this you have a few options. Generally, you are probably going to convert your colors to HSV, which is "Hue Saturation Value." This can help you actually pinpoint a more specific color, based on hue and saturation ranges, with a variance of value, for example. If you wanted, you could actually produce filters based on BGR values, but this would be a bit more difficult. If you're having a hard time visualizing HSV, don't feel silly, check out the Wikipedia page on HSV, there is a very useful graphic there for you to visualize it. Hue for color, saturation for the strength of the color, and value for light is how I would best describe it personally.

Now let's hop in.

```
import cv2
importnumpyasnp

cap= cv2.VideoCapture(0)

while(1):
    _, frame =cap.read()
hsv= cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

lower_red=np.array([30,150,50])
upper_red=np.array([255,255,180])

mask= cv2.inRange(hsv,lower_red,upper_red)
res= cv2.bitwise_and(frame,frame, mask= mask)

cv2.imshow('frame',frame)
cv2.imshow('mask',mask)
cv2.imshow('res',res)

    k =cv2.waitKey(5)&0xFF
if k ==27:
break

cv2.destroyAllWindows()
cap.release()
```

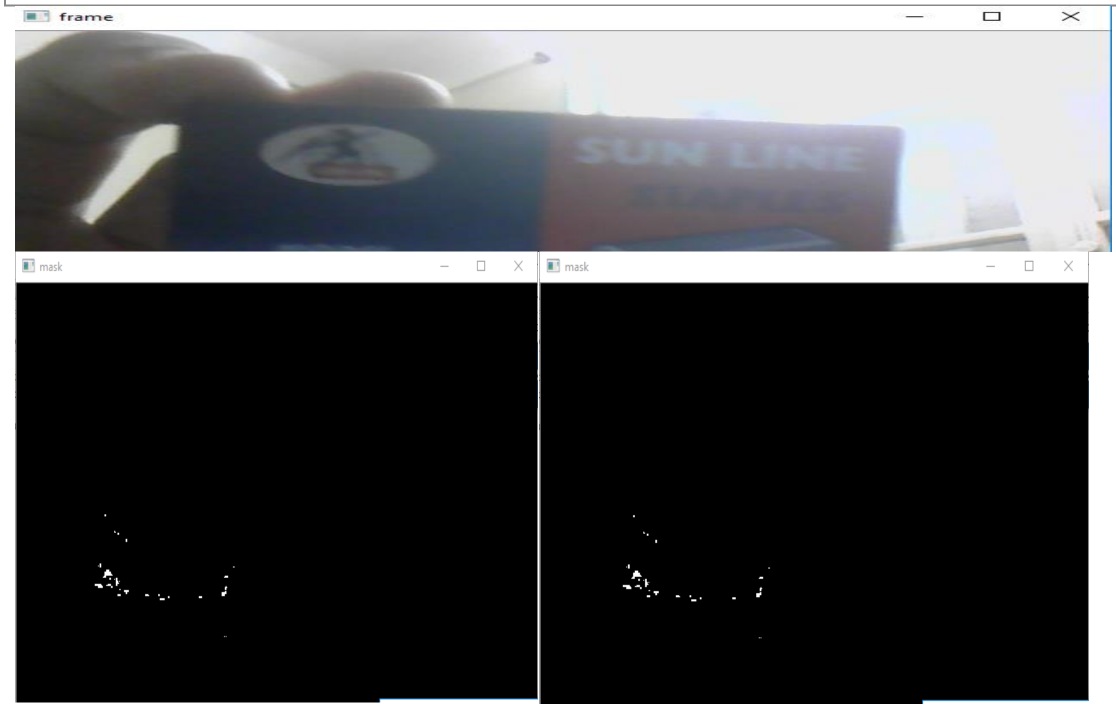


Figure 2.1 the result of color filture in real time

This is just an example, with reds as the target. The way this works is what we see will be anything that is between our ranges here, basically 30-255, 150-255, and 50-180. This is a for a red, but feel free to try to find your own colors. The reason why HSV works best here is because we want a range of colors, and we generally want the same-ish color in this case. Many times, a typical red will still have some green and blue, so we would have to allow for some green and some blue, but then we'd want pretty much full red. This means we'd get lower-light mixes of all colors still at this point.

7.6 Creating your own HaarCascadeOpenCVPython[05]

when you want to build a Haar Cascade, you need "positive" images, and "negative" images. The "positive" images are images that contain the object you want to find. This can either be images that just mainly have the object, or it can be images that contain the object, and you specify the ROI (region of interest) where the object is. With these positives, we build a vector file that is basically all of these positives put together. One nice thing about the positives is that you can actually just have one image of the object you wish to detect, and then have a few thousand negative images. Yes, a few thousand. The negative images can be anything, except they cannot contain your object.

From here, with your single positive image, you can use the `opencv_createsamples` command to actually create a bunch of positive examples, using your negative images. Your positive image will be superimposed on these negatives, and it will be angled and all sorts of things. It actually can work pretty well, especially if you are really just looking for one specific object. If you are looking to identify all screwdrivers, however, you will want to have thousands of unique images of screwdrivers, rather than using the `opencv_createsamples` to generate samples for you. We'll keep it simple and just use one positive image, and then create a bunch of samples with our negatives.

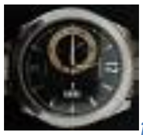


*figure 2.2 watch in size 50*50*

Here's another scenario where you will likely enjoy this better if you use your own image. If things go wrong, try with mine and see where maybe you went wrong, but I suggest you take your own picture. Keep it small. 50x50 pixels is pushing it. Ok great, getting a positive image is no problem! There is just one problem. We need thousands of negative images. Possibly in the future, we may want thousands of positive images too. Where in the world can we do that? There's quite a useful site, based on the concept of WordNet, called ImageNet. From here, you can find images of just about anything. In our case, we want watches, so search for watches, and you will find tons of categories of watches. Let's go with analog watches. Awesome! It gets better though. Check out that downloads tab! There's a URL for all of the analog watches!. Okay but I said we will just use the one positive, so we just detect the one watch. If you want to detect "all" watches, prepare to get more like 50,000 images of watches, and at least 25,000 "negative" images.

After that, prepare to have quite the server, unless you want your Haar Cascade training to take a week. So how might we get negatives? The whole point of ImageNet is for image training, so their images are pretty specific. Thus, if we search for people, cars, boats, planes...whatever, chances are, there will be not watches. You might see some watches on people or something like that, but you get the idea . we have all these images, first, we actually want all of these to be the same size, and a whole lot smaller! Gosh if only we knew of a way to manipulate images... hmm... Oh right this is an OpenCV tutorial! We can probably handle it. So, first, what we're going to do here is write a quick script that will visit these URL lists, grab the links, visit the links, pull the images, resize them, save them, and repeat until we're done. When our directories are full of images, we also need a sort of description file that describes the images. For positives, this file is a massive pain to create manually, since you need to specify the exact Region of Interest for your object, per image. Gross. Luckily the create_samples method places the image randomly and does all that work for us. We just need a simple descriptor for the negatives, but that's no problem, we can do that while we pull and manipulate the images.

1- Download the pic (data acquisition)

```
importurllib.request
import cv2
importnumpyasnp
importos

defstore_raw_images():
neg_images_link='//image-
net.org/api/text/imagenet.synset.geturls?wnid=n00523513'
neg_image_urls=urllib.request.urlopen(neg_images_link).read().decod
e()
pic_num=1

ifnotos.path.exists('neg'):
os.makedirs('neg')

for i inneg_image_urls.split('\n'):
try:
print(i)
urllib.request.urlretrieve(i,"neg/"+str(pic_num)+".jpg")
img= cv2.imread("neg/"+str(pic_num)+".jpg",cv2.IMREAD_GRAYSCALE)
# should be larger than samples / pos pic (so we can place our
image on it)
resized_image=cv2.resize(img,(100,100))
cv2.imwrite("neg/"+str(pic_num)+".jpg",resized_image)
pic_num+=1

exceptExceptionas e:
print(str(e))
```

Simple enough, this script will visit the links, grab the URLs, and proceed to visit them. From here, we grab the image, convert to grayscale, resize it, then save it. We use a simple counter for naming the images. Go ahead and run it. As you can probably see, there are a lot of missing pictures and such. That's okay. More problematic is some of these error pictures. Basically all white with some text that says they are no longer available, rather than serving and HTTP error. Now, we have a couple choices. We can just ignore this, or fix it. Hey, it's an image without a watch, so whatever right? Sure, you could take that opinion, but if you use this pulling method for positive then this is definitely a problem. You could manually delete them... or we can just use our new Image Analysis knowledge to detect these silly images and remove them!

I went ahead and made a new directory, calling it "uglies." Within that directory, I have click and dragged all ugly image versions (just one of each). There's only one major offender that I found with the negatives, so I just have one.

2- Let's write a script to find all instances of this image and delete it.

```
deffind_uglies():
match=False
forfile_typein['neg']:
forimginos.listdir(file_type):
for ugly inos.listdir('uglies'):
try:
current_image_path=str(file_type)+'/'+str(img)
ugly= cv2.imread('uglies/'+str(ugly))
question= cv2.imread(current_image_path)
ifugly.shape==question.shapeandnot(np.bitwise_xor(ugly,question).an
y()):
print('That is one ugly pic! Deleting!')
print(current_image_path)
os.remove(current_image_path)
exceptExceptionas e:
print(str(e))
```

3- The last step is we need to create the descriptor file for these negative images

```
defcreate_pos_n_neg():
forfile_typein['neg']:

forimginos.listdir(file_type):

iffile_type=='pos':
line=file_type+'/'+img+' 1 0 0 50 50\n'
with open('info.dat','a')as f:
f.write(line)
eliffile_type=='neg':
line=file_type+'/'+img+'\n'
with open('bg.txt','a')as f:
f.write(line)
```

Run that, and you have a bg.txt file. Now, I understand some of you may not have the best internet connection, so I will be a good guy Greg and upload the negative images. Alright, so we decided we're going to just use the one image for the postive foreground image. Thus, we need to do create_samples. This means, we need to move our neg directory and the bg.txt file to our server. If you ran all this code on your server, don't worry about it.

4- We're ready to create some positive samples now, based on the watch5050.jpg image. To do this, run the following via the terminal

```
opencv_createsamples -img watch5050.jpg -bg bg.txt -info
info/info.lst -pngoutput info -maxxangle 0.5 -maxyangle 0.5
-maxzangle 0.5 -num 1950
```

5- we now need to create the vector file

```
opencv_createsamples -info info/info.lst -num 1950 -w 20 -h
20 -vecpositives.vec
```

6- Now let's run the train command

```
opencv_traincascade -data data -vecpositives.vec -bg bg.txt
-numPos 1800 -numNeg 900 -numStages 10 -w 20 -h 20
```

In result of this steps we have the xml to watchesHaarCascade

7- Testing the xml file with python and open cv.show Figure 2.3

```
importnumpyasnp
import cv2

face_cascade=cv2.CascadeClassifier('haarcascade_frontalface_defau
lt.xml')
eye_cascade=cv2.CascadeClassifier('haarcascade_eye.xml')

#this is the cascade we just made. Call what you want
watch_cascade=cv2.CascadeClassifier('watchcascade10stage.xml')

cap= cv2.VideoCapture(0)

while1:
ret,img=cap.read()
gray= cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
faces=face_cascade.detectMultiScale(gray,1.3,5)
```

```
# add this

# image, reject levels level weights.
watches=watch_cascade.detectMultiScale(gray,50,50)

# add this
for(x,y,w,h)in watches:
cv2.rectangle(img,(x,y),(x+w,y+h),(255,255,0),2)

for(x,y,w,h)in faces:
cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)


roi_gray= gray[y:y+h, x:x+w]
roi_color=img[y:y+h, x:x+w]
eyes=eye_cascade.detectMultiScale(roi_gray)
for(ex,ey,ew,eh)in eyes:

cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)

cv2.imshow('img',img)
    k =cv2.waitKey(30)&0xff
if k ==27:
break

cap.release()
cv2.destroyAllWindows()
```

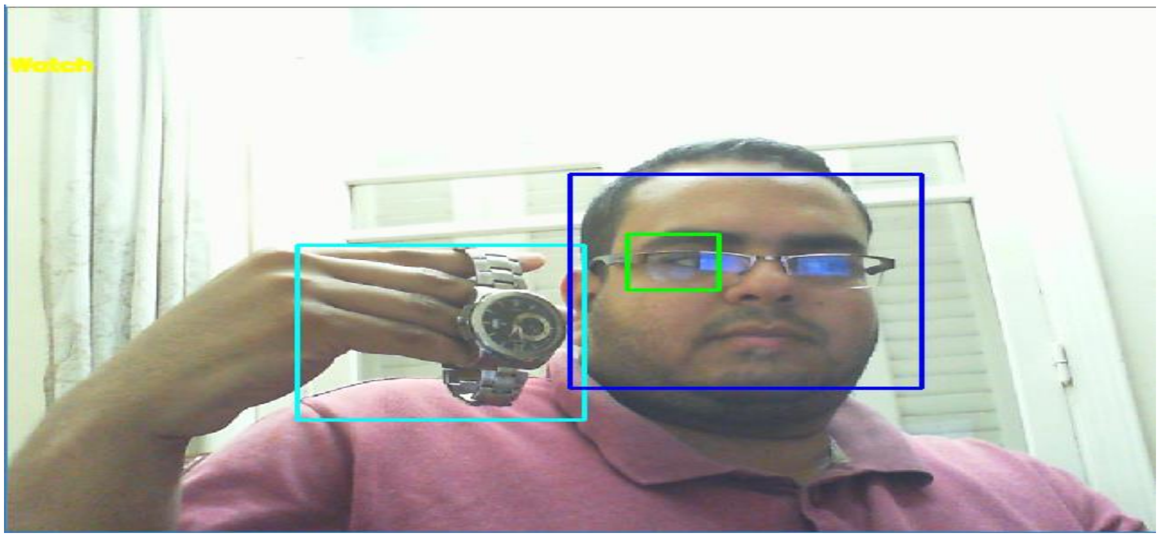


Figure 2.3 the test result of xml

8 Voice commend

Voice commend is a controlled by means of the human voice. By removing the need to use buttons, dials and switches, consumers can easily operate appliances with their hands full or while doing other tasks.

8.1 TTS

Text to speech done by local server called espeak

8.1.1 Espeak

a compact open source software speech synthesizer for Linux, Windows and other platforms. It uses a formant synthesis method, providing many languages in a small size. Much of the programming for eSpeakNG's language support is done using rule files with feedback from native speakers

8.1.1.1 Test

```
espeak-ng -v en "Hello [[w3:ld]]"
```

8.2 STT

Speech to Text done by using google cloud API (speech to text) with shell program

APPENDIX A

8.3 PROCCESING

Using the python program to select some word from text and take the control the robot

```
while True:
subprocess.call(['./speech-rec.sh'])
   with open('MyText.txt', 'r') as f:
lineArr=f.read().split()
   if 'left' in lineArr:
      print ("left")
           subprocess.call(["espeak","-s140 -ven+18 -z","i will turn to left"])
      left()
elif 'right' in lineArr :
      print ("right")
           subprocess.call(["espeak","-s140 -ven+18 -z","i will turn to right"])
      right()
```

```
elif 'forward' in lineArr :
        print ("forward")
              subprocess.call(["espeak","-s140 -ven+18 -z","i will go to forward"])
              forward()
elif 'back' in lineArr :
    print ("back")
              subprocess.call(["espeak","-s140 -ven+18 -z","i will go to back"])
        backward()
elif 'stop' in lineArr :
        print ("stop")
              subprocess.call(["espeak","-s140 -ven+18 -z","i will stop now"])
        stop()
f.close()
    with open('MyText.txt', 'w') as f:
f.write(" ")
f.close()
```

9 Webiopi server (iot)[06]

- **Control, debug, and use your Pi's GPIO, sensors and converters from a web browser or any app**
- **WebIOPi is the perfect Swiss-knife to make connected things**
- **Developed and provided by Eric PTAK (trouch)**
- **Runs on** Raspberry Pi

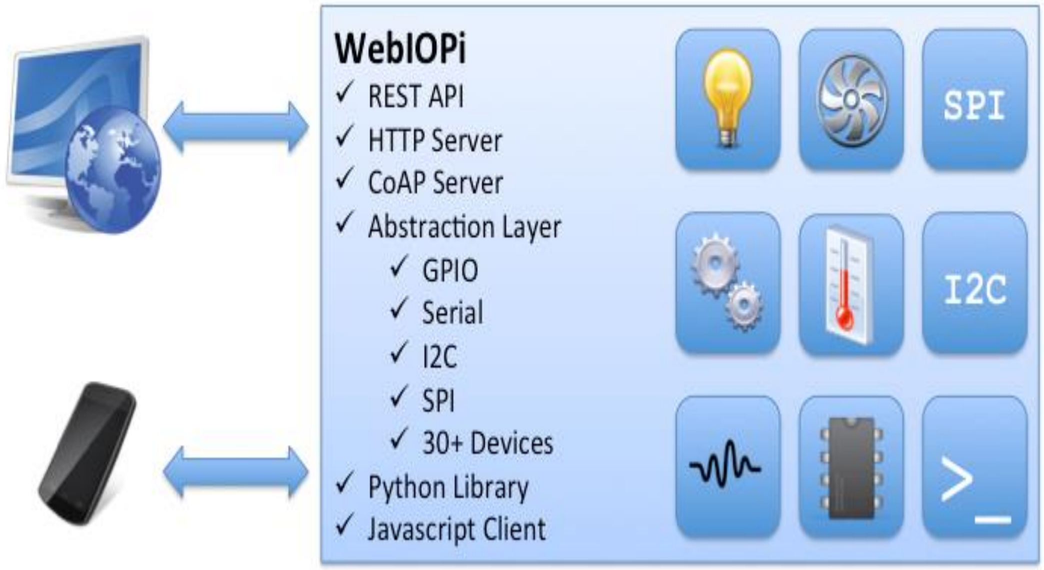


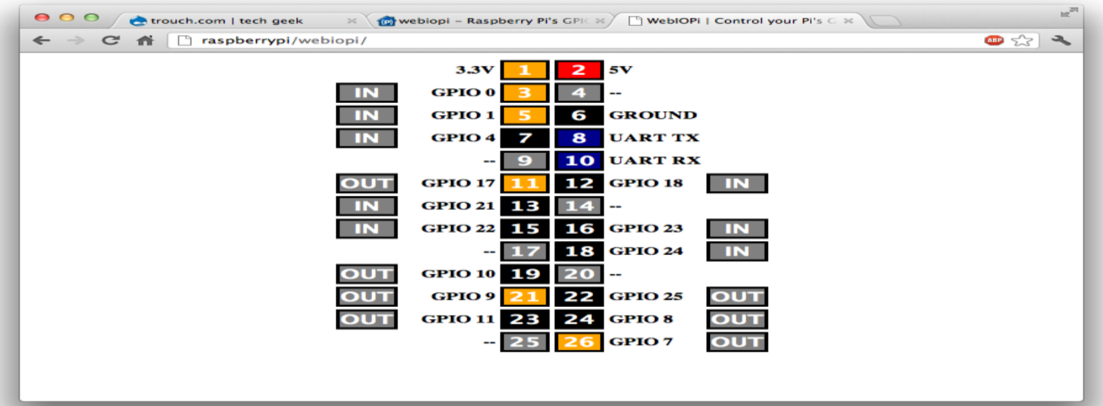
Figure 4.1 webiopi service

9.1 Features

- Written in **Python**, with facilities to load and execute custom script, using a comprehensive structure with **setup and loop functions**
- Unified **Serial/SPI/I2C support** with a complete and **consistent set of functions** to control more than 30 devices, including most used analog converters, I/O expander and sensors
- **Javascript/HTML client** library to make Web UI
- **Python/Java clients**, to make **Pi-to-Pi** systems or Android applications
- **CoAP support** brings the best Internet of Things protocol on the Pi, as a future proof of Pi possibilities
- Includes simple web apps, to debug GPIO, devices and Serial interface

9.2 GPIO header web app

The GPIO header web application is included to quickly debug and controls GPIO.

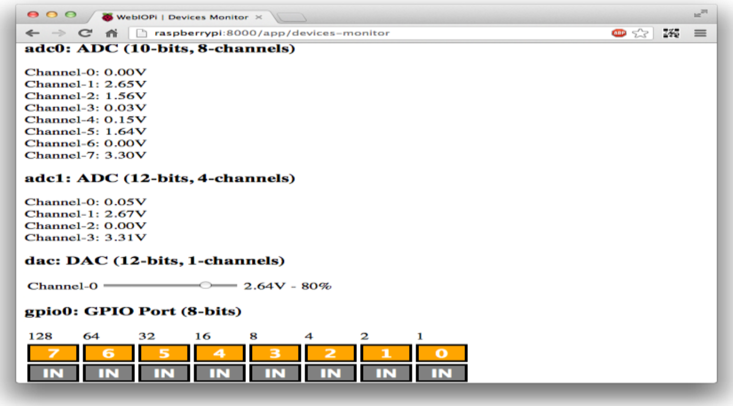


	3.3V	1	2	5V	
IN	GPIO 0	3	4	--	
IN	GPIO 1	5	6	GROUND	
IN	GPIO 4	7	8	UART TX	
	--	9	10	UART RX	
OUT	GPIO 17	11	12	GPIO 18	IN
IN	GPIO 21	13	14	--	
IN	GPIO 22	15	16	GPIO 23	IN
	--	17	18	GPIO 24	IN
OUT	GPIO 10	19	20	--	
OUT	GPIO 9	21	22	GPIO 25	OUT
OUT	GPIO 11	23	24	GPIO 8	OUT
	--	25	26	GPIO 7	OUT

Figure 4.2 default page of webiopi (GPIO state)[adopted]

9.3 Device monitor web app

The device monitor allows to debug and controls converters or sensors plugged in the GPIO/SPI/I2C...


WebIOPi | Devices Monitor
raspberrypi:8000/app/devices-monitor

```
adc0: ADC (10-bits, 8-channels)

Channel-0: 0.00V
Channel-1: 2.65V
Channel-2: 1.56V
Channel-3: 0.03V
Channel-4: 0.15V
Channel-5: 1.64V
Channel-6: 0.00V
Channel-7: 3.30V

adc1: ADC (12-bits, 4-channels)

Channel-0: 0.05V
Channel-1: 2.67V
Channel-2: 0.00V
Channel-3: 3.31V

dac: DAC (12-bits, 1-channels)

Channel-0                2.64V - 80%

gpio0: GPIO Port (8-bits)

128   64    32    16    8     4     2     1
 7     6     5     4     3     2     1     0
 IN    IN    IN    IN    IN    IN    IN    IN
```

Figure 4.3 device monitor [adopted]

9.4 Framework basis

9.4.1 Light Control with auto on/off

- Making an IoT application using WebIOPi is very easy once we catch how it works and what is necessary.
- WebIOPi includes an HTTP server that provides both HTML resources and a REST API to control things. Your browser will first load a HTML file, then the included Javascript will make Asynchronous calls to the REST API to control and update the UI. This method is very efficient, because it don't need to refresh and download the whole page.

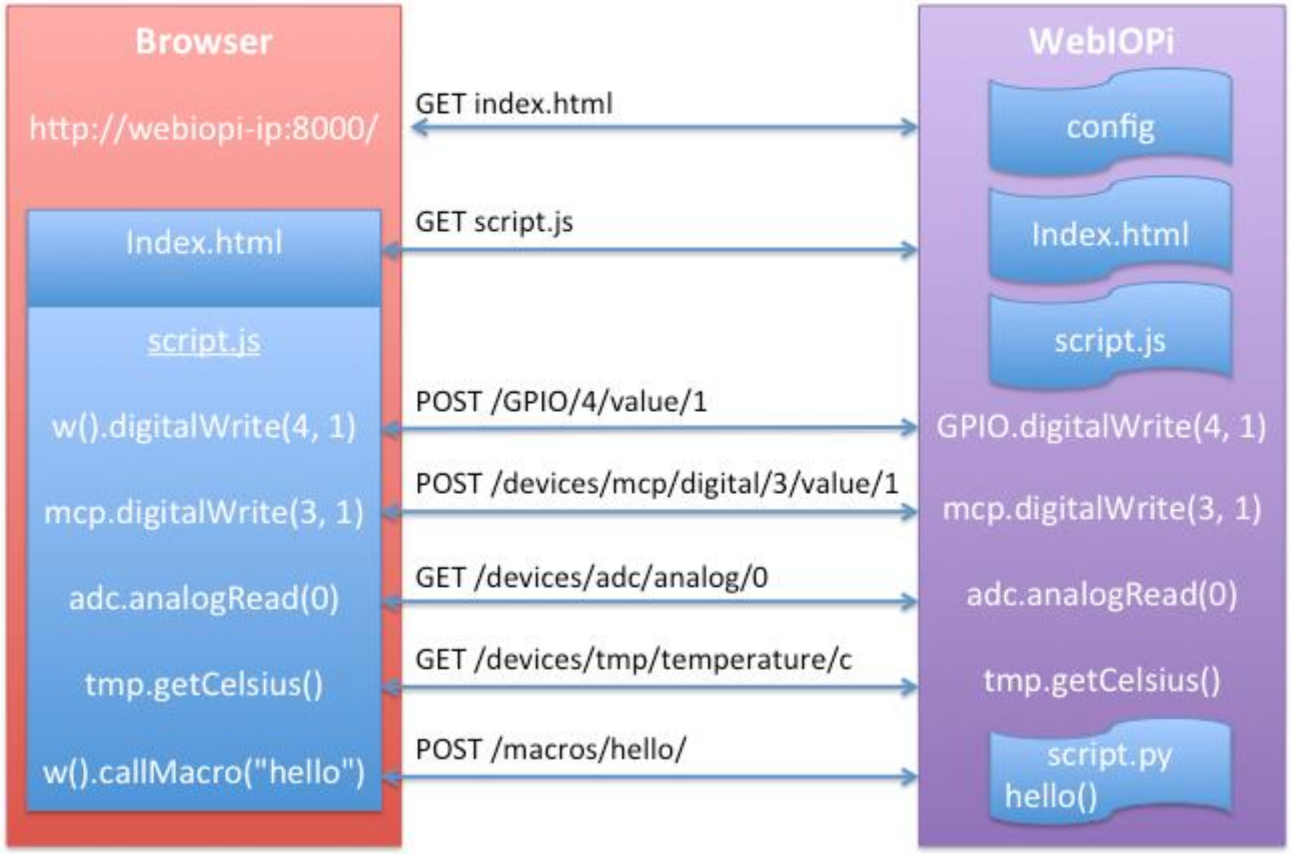


Figure 4.4 basic frame work [adopted]

- You can build your HTML / Web UI from scratch, using the WebIOPi JS library or not.
- You can extend the WebIOPi behavior by loading custom Python script using an Arduino like syntax with setup/loop functions

Assume we simply want a button to control a light, which will be automatically turned on and off. I say a light, but it can be a anything you want. The customization process has 3 steps :

1. Write a simple Python script to initialize the GPIO and handle auto on/off
2. Write a simple HTML/Javascript page
3. Configure WebIOPi Server

9.4.1.1 Prerequisites

1. If not already done, install WebIOPi on your Raspberry Pi.
2. Create a folder somewhere on your Pi, for instance **/home/pi/myproject**. *This is the main folder.*
3. Create another **python** folder in the folder previously created. *We will store Python script file here.*
4. Create another **html** folder next to **python**. *We will store HTML and other ressources here.*

You should have something like this :

```
home
-pi
  -myproject
    -python
    -html
```

9.4.1.2 Python script

Create a **script.py** file in your **python** folder, eg. **/home/pi/myproject/python/script.py**. This file will be loaded and executed by the WebIOPi server.

```
importwebiopi
importdatetime

GPIO = webiopi.GPIO

LIGHT = 17 # GPIO pin using BCM numbering

HOUR_ON  = 8  # Turn Light ON at 08:00
HOUR_OFF = 18 # Turn Light OFF at 18:00

# setup function is automatically called at WebIOPi startup
def setup():
    # set the GPIO used by the light tooutput
GPIO.setFunction(LIGHT, GPIO.OUT)
```

```
    # retrieve currentdatetime
now = datetime.datetime.now()

    # testif we arebetweenONtimeandtun the light ON
if ((now.hour>= HOUR_ON) and (now.hour< HOUR_OFF)):
GPIO.digitalWrite(LIGHT, GPIO.HIGH)

# loopfunctionis repeatedly called byWebIOPi
defloop():
    # retrieve currentdatetime
now = datetime.datetime.now()

    # toggle light ON all daysat the correct time
if ((now.hour == HOUR_ON) and (now.minute == 0) and (now.second == 0)):
if (GPIO.digitalRead(LIGHT) == GPIO.LOW):
GPIO.digitalWrite(LIGHT, GPIO.HIGH)

    # toggle light OFF
if ((now.hour == HOUR_OFF) and (now.minute == 0) and (now.second == 0)):
if (GPIO.digitalRead(LIGHT) == GPIO.HIGH):
GPIO.digitalWrite(LIGHT, GPIO.LOW)

    # gives CPU sometimebefore looping again
webiopi.sleep(1)

# destroy functionis called atWebIOPishutdown
def destroy():
GPIO.digitalWrite(LIGHT, GPIO.LOW)
```

That's it ! No HTTP boring stuff. The file will be loaded by the WebIOPiserver which already handle remote control.

9.4.1.3 HTML/Javascript

Create a **index.html** file in your **html** folder, eg. **/home/pi/myproject/html/index.html**. The WebIOPiJavascript library allows you to make your own interface easily with buttons bound to GPIO. You only need a single <script> tag to include **/webiopi.js**. It will then automatically load jQuery, a nice JS library.

You don't need to put webiopi.js and jquery.js in your project html folder. You just need the <script> *tag in your index.html. The WebIOPi server filters browsers requests to serves both webiopi.js and jquery.js files from the default WebIOPi resource folder.*

The index.html is composed of few HTML tags, including a little Javascript part and few CSS lines. The most important to take care about is the anonymous Javascript function passed to WebIOPi JS library with **webiopi().ready()**. This ensureWebIOPi and jQuery libraries are loaded before modifying the UI. Also take take of the parenthesis when prefixing functions calls with **webiopi()**. After the button creation, we use a jQuery function to append it to a HTML element, declared later in the <body> tag.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<metahttp-equiv="Content-Type"content="text/html; charset=UTF-8">

<title>WebIOPi | Light Control</title>

<scripttype="text/javascript"src="/webiopi.js"></script>

<scripttype="text/javascript">

webiopi().ready(function() {

// Create a "Light" labeled button for GPIO 17

var button = webiopi().createGPIOButton(17, "Light");


// Append button to HTML element with ID="controls" using jQuery

        $("#controls").append(button);


// Refresh GPIO buttons

// pass true to refresh repeatedly of false to refresh once

webiopi().refreshGPIO(true);

    });

</script>

<styletype="text/css">
```

```
button {
display: block;
margin: 5px5px5px5px;
width: 160px;
height: 45px;
font-size: 24pt;
font-weight: bold;
color: white;
        }

#gpio17.LOW {
background-color: Black;
        }

#gpio17.HIGH {
background-color: Blue;
        }
</style>
</head>
<body>
<divid="controls"align="center"></div>
</body>
</html>
```

9.4.1.4 Configuration

Edit **/etc/webiopi/config** :

- Locate [SCRIPTS] section, add following line to load your Python script

```
...
[SCRIPTS]
myproject = /home/pi/myproject/python/script.py
...
```

- Locate [HTTP] section, add following line to tell WebIOPi where to find your HTML resources

```
...
[HTTP]
doc-root = /home/pi/myproject/html
...
```

The two steps above are enough to run our app, but we want to limit remote control to the Light only. By default, we can remotely change function and value of all GPIO.

- Locate [REST] section, add following lines

```
...
[REST]
gpio-export = 17
gpio-post-value = true
gpio-post-function = false
...
```

- **gpio-export** limits GPIO exported on the REST API (using BCM numbering)
- **gpio-post-value** allows/forbids to remotely change GPIO values (LOW/HIGH)
- **gpio-post-function** allows/forbids to remotely change GPIO function (IN/OUT)

9.4.1.5 Testing

To debug your script and config, you should first run WebIOPi foreground before using the daemon service :

```
sudowebiopi -d -c /etc/webiopi/config
```

You can now open your browser to your Raspberry Pi IP at port 8000 (http://raspberrypi:8000/) and control a light/led connected to GPIO 17

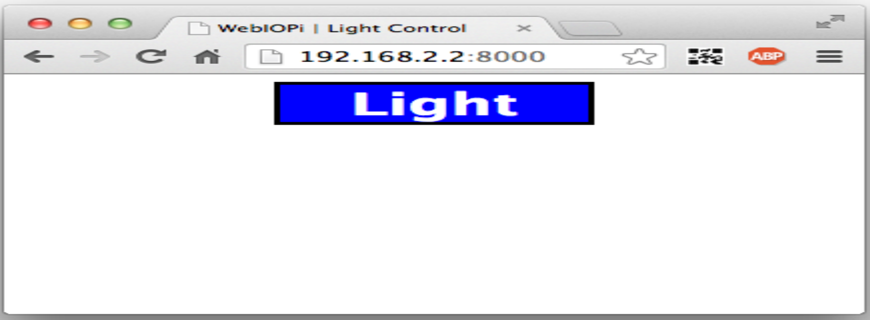
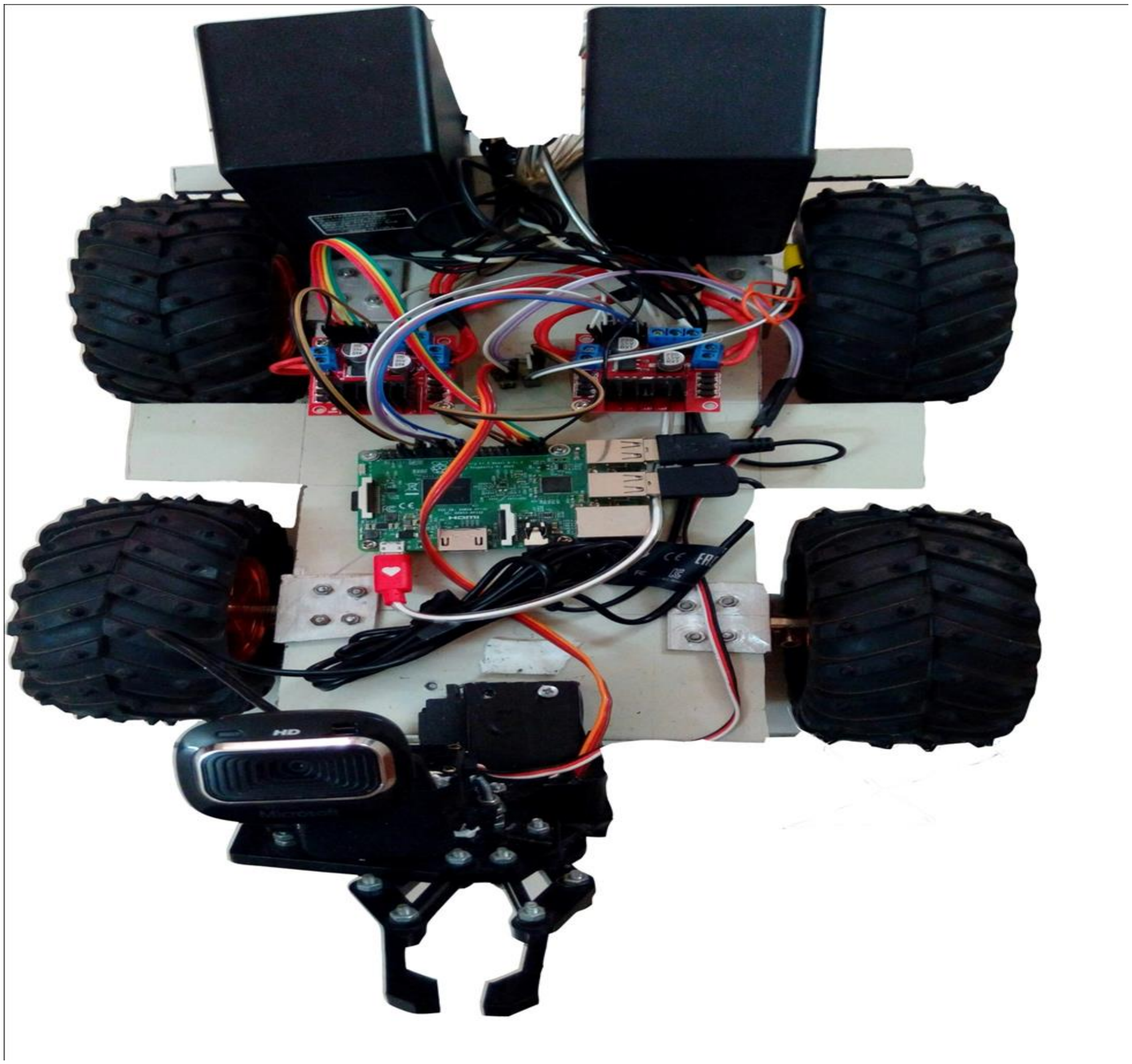


figure4.5 the result of code

9.5 weaved server

using for connette local server webiopi with the public internet through http protcol

10 prototype design



10.1 Body (aluminium cladding sheet)

Aluminium composite panel (ACP), also aluminium composite material (ACM), is a type of flat panel that consists of two thin aluminium sheets bonded to a non-aluminium core. ACPs are frequently used for external cladding or facades of buildings, insulation, and signage.[07] If the core material is flammable, usage may be problematic as a building material and some jurisdictions have banned their use

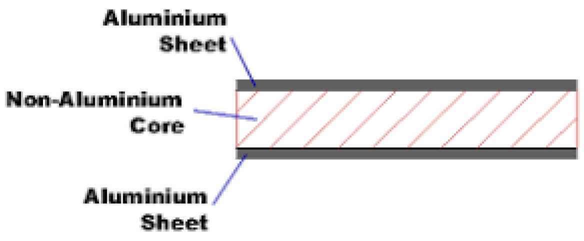


Figure 10.1 aluminium cladding sheet [adopted][07]

10.2 Links

We using aluminium sheet show figure 10.2

Figure 10.2 aluminium link

10.3 Drivers

10.3.1 Introduction

[08]The driver modules are based on L298N H-bridge, a high current, high voltage dual full bridge driver manufactured by ST Company. It can drive up to 2 DC motors 2A each. It can also drive one stepper motor or 2 solenoids.Thedriver can control both motor RPM and direction of rotation. The RPM is controlled using PWM input to ENA or ENB pins, while of rotation direction is controlled by supplying high and low signal to EN1-EN2 for the first motor or EN3-EN4 for second motor. This Dual H-Bridge driver is capable of driving voltages up to 46V.

10.3.2 Features

- Dual H bridge drive (can drive 2 DC motors)
- Chip L298N
- Logical voltage 5V
- Drive voltage 5V-35V
- Logic current 0mA-36mA
- Drive current 2A(For each DC motor))
- Weight 30g
- Size: 43*43*27mm

10.3.3 Consist of

- DC motor 1 “+” or stepper motor A+
- DC motor 1 “-” or stepper motor A-
- 12V jumper – remove this if using a supply voltage greater than 12V DC. This enables power to the onboard 5V regulator
- Connect your motor supply voltage here, maximum of 35V DC. Remove 12V jumper if >12V DC
- GND
- 5V output if 12V jumper in place, ideal for powering your Arduino (etc)
- DC motor 1 enable jumper. Leave this in place when using a stepper motor. Connect to PWM output for DC motor speed control.
- IN1 ,IN2 ,IN3 ,IN4
- DC motor 2 enable jumper. Leave this in place when using a stepper motor. Connect to PWM output for DC motor speed control
- DC motor 2 “+” or stepper motor B+
- DC motor 2 “-” or stepper motor B-

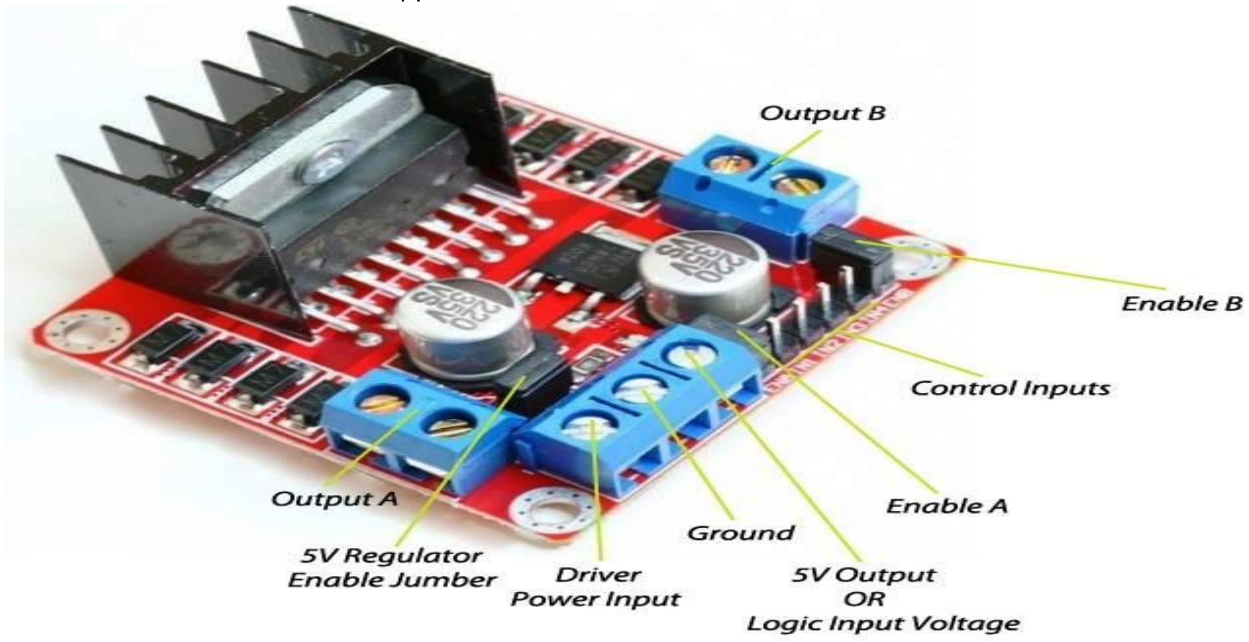


Figure 10.3 motor driver [adopted][08]

10.4 Motors

10.4.1 Introduction

[09]This is 12V DC motor, almost double the power of our 12V DC motor. It is a powerful motor fitted with a 34:1 metal gearbox. The gears are all steel and the output shaft is 4mm diameter. The motor has extra rear shaft (on the high speed side, before the gearbox) which allow you to mount motor encoder for RPM counting and superior control.

10.4.2 Feature

- Voltage = 12V
- No load current = 250mA
- Stall Current = 3.3 A
- Stall Torque = 8.8 Kg/cm
- Motor Rated RPM = 250
- Shaft Diameter = 4 mm
- Motor Length (without shaft) = 52 mm
- Motor Diameter = 25 mm

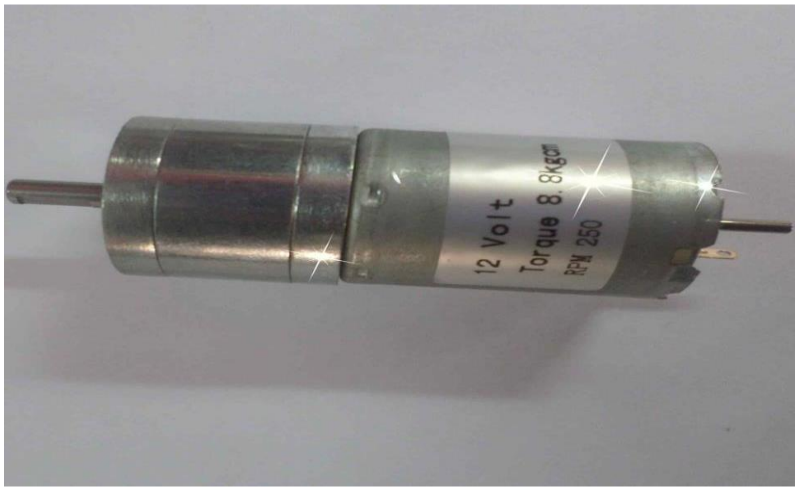


Figure 10.4 dc motor[adopted][09]

10.5 Wheels

10.5.1 Introduction

[10]This rugged, 4 -wheel-drive chassis from Danu Electronics is designed to excel at traversing rough terrain and steep inclines, making it a great platform for any robot that needs to perform tasks in a complex outdoor environment.

10.5.2 Features

- four powerful DC motors with brass brushes
- 34:1 or 75:1 steel gearboxes that drive large (120mm diameter) spiked tires,
- A unique “super-twist” suspension system acts to keep each wheel in contact with the ground for maximum traction, even when driving over uneven or bumpy surfaces.
- The suspension can be adjusted to suit different loads and conditions.
- The chassis is made from a 2mm-thick corrosion-resistant anodized aluminum plate,
- All of the nuts, bolts, and screws are stainless steel, and the brass fittings and suspension springs are nickel-plated.

10.5.3 Specifications

- Size: 420 × 300 × 130 mm (16.5" × 12" × 5")
- Weight: 2.7 kg (6.0 lb)
- Ground clearance: 60 mm (2.5") when lightly loaded
- Maximum recommended payload: 5 kg (11 lb)
- Recommended motor voltage: 2 – 7.5 V
- Stall current at 7.2 V: 6.6 A per motor
- No-load current at 7.2 V: 420 am per motor
- No-load output shaft speed at 7.2 V:
- 350 RPM for the version with 34:1 gearboxes
- 160 RPM for the version with 75:1 gearboxes
- Stall torque at 7.2 V:
- 5 kg-cm (70 oz-in) per motor for the version with 34:1 gearboxes
- 11 kg-cm (160 oz-in) per motor for the version with 75:1 gearboxes

Figure 10.4 wheel drive[adopted][10]

10.6 Raspberry pi 3 [11]

10.6.1 Feature

- 1.2GHz 64-bit quad-core ARM Cortex-A53 CPU (BCM2837)
- 1GB RAM (LPDDR2 SDRAM)
- On-board Wireless LAN - 2.4 GHz 802.11 b/g/n (BCM43438)
- On-board Bluetooth 4.1 + HS Low-energy (BLE) (BCM43438)
- 4 x USB 2.0 ports
- 10/100 Ethernet
- 40 GPIO pins
- Full size HDMI 1.3a port
- Combined 3.5mm analog audio and composite video jack
- Camera interface (CSI)
- Display interface (DSI)
- Micros slot
- Video Core IV multimedia/3D graphics core @ 400MHz/300MHz
- With the ARMv8 processor it can run the full range of ARM GNU/Linux distributions, including Snappy Bunt Core, as well as Microsoft Windows 10 IoT edition.

10.6.2 Performance

Compared with the Raspberry Pi 2 the CPU is running at a 33% faster clock rate (1.2GHz vs. 0.9GHz) the more modern core also means a more efficient instruction set, especially when performing operations on 64-bit values.Video and 3D performance has also seen a bump with the Video Core being clocked at 400MHz for video processing (up from 250MHz) and the 3D graphics processor running at 300MHz (up from 250MHz).In general use you can expect the Raspberry Pi 3 to perform around 50% quicker than the Raspberry Pi 2 running the same software. In the future as software adds optimizations for the architecture the gap will widen further.Wireless LAN, Bluetooth, The biggest change is the inclusion of integrated Wireless LAN and Bluetooth 4.1 on board adding two great connectivity options straight out of the box. The Wireless LAN should be compatible with all home network setups and is supported in the latest NOOBs image.We've achieved speeds of around 40Mbit/sec using the new built in adaptor which is similar to the speeds we see with USB Wi-Fi dongles in the same setup.Bluetooth will allow you to use a wireless keyboard/track pad without extra dongles, keeping things nice and tidy.

10.6.3 GPIO and layout

The GPIO header and layout remains the same as before so all of your existing HATs and add-on boards will work fine with the Raspberry Pi 3.
Some cases will not be ideal:

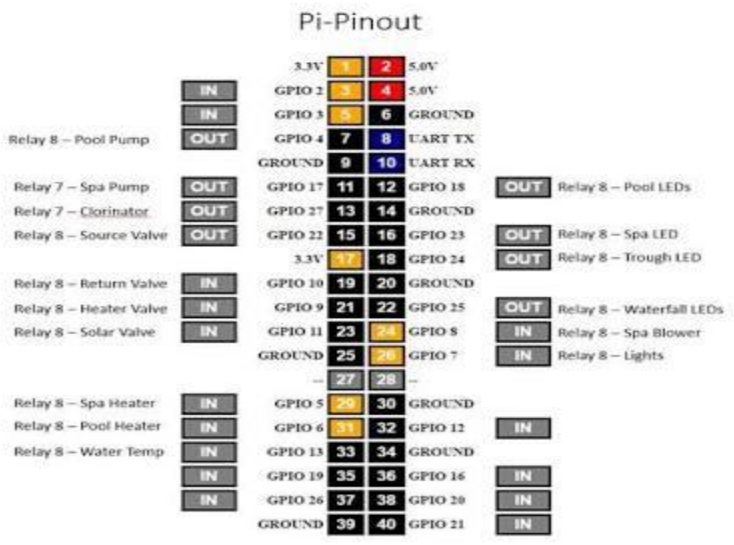

Pi-Pinout
3.3V 1 2 5.0V
IN GPIO 2 3 4 5.0V
IN GPIO 3 5 6 GROUND
Relay 8 – Pool Pump OUT GPIO 4 7 8 UART TX
GROUND 9 10 UART RX
Relay 7 – Spa Pump OUT GPIO 17 11 12 GPIO 18 OUT Relay 8 – Pool LEDs
Relay 7 – Clorinator OUT GPIO 27 13 14 GROUND
Relay 8 – Source Valve OUT GPIO 22 15 16 GPIO 23 OUT Relay 8 – Spa LED
3.3V 17 18 GPIO 24 OUT Relay 8 – Trough LED
Relay 8 – Return Valve IN GPIO 10 19 20 GROUND
Relay 8 – Heater Valve IN GPIO 9 21 22 GPIO 25 OUT Relay 8 – Waterfall LEDs
Relay 8 – Solar Valve IN GPIO 11 23 24 GPIO 8 IN Relay 8 – Spa Blower
GROUND 25 26 GPIO 7 IN Relay 8 – Lights
– 27 28 –
Relay 8 – Spa Heater IN GPIO 5 29 30 GROUND
Relay 8 – Pool Heater IN GPIO 6 31 32 GPIO 12 IN
Relay 8 – Water Temp IN GPIO 13 33 34 GROUND
IN GPIO 19 35 36 GPIO 16 IN
IN GPIO 26 37 38 GPIO 20 IN
GROUND 39 40 GPIO 21 IN

Figure 10.5 gpio for rpi 3[adopted][11]

10.6.4 Power of raspberry pi

We used separate phone charger with 2.1 am to keep it stable
The Raspberry Pi 3 is powered by a +5.1V micro USB supply. Exactly how much current (am) the Raspberry Pi requires is dependent on what you connect to it. We have found that purchasing a 2.5A power supply from a reputable retailer will provide you with ample power to run your Raspberry Pi. You can purchase the official Raspberry Pi Power Supply from our website, and you can learn more about the differing power requirements of the various models of the Raspberry Pi on our FAQ page.Typically, the model B uses between 700-1000mA depending on what peripherals are connected; the model A can use as little as 500mA with no peripherals attached. The maximum power the Raspberry Pi can use is 1 Amp. If you need to connect a USB device that will take the power requirements above 1 Amp, then you must connect it to an externally-powered USB hub.The power requirements of the Raspberry Pi increase as you make use of the various interfaces on the Raspberry Pi. The GPIO pins can draw 50mA safely, distributed across all the pins; an individual GPIO pin can only safely draw 16mA. The HDMI port uses 50mA, the camera module requires 250mA, and keyboards and mice can take as little as 100mA or over 1000mA

10.7 Camera & microphone

[12]We used Microsoft hd3000 camera with internal microphone

10.7.1 Features

- Widescreen with 720p HD video chat and recording
- Noise reducing microphone
- True Color Technology for bright and colorful video Dimensions
- Length: 4.3" / 109mm
- Width: 1.75" / 44.5mm
- Crystal-clear audio and built-in unidirectional microphone with acoustic noise cancellation.
- Universal attachment base
- Features
- Widescreen with 720p HD video chat and recording
- True Color Technology for bright and colorful video

Figure 10.6 microsoft webcam 3000hd[adopted][12]

10.8 Gripper

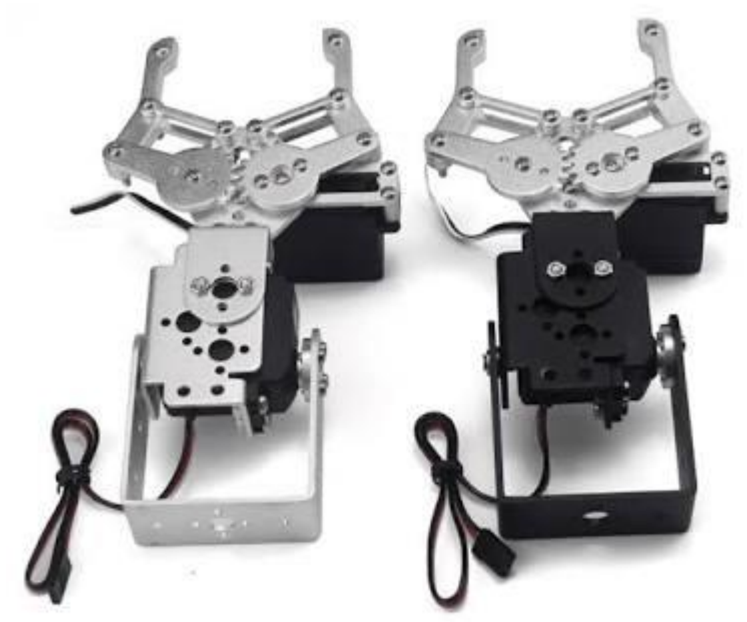


Figure 10.7 The robot includes 2 servos with rigid mechanical structure [adopted][13]

10.8.1 Features

- Made from 3mm thick aluminum sheet
- Servos
- Portable (Light weight)
- Servo Features
- Standard PWM Interface
- Operating Voltage 4.8V-6V

11 Work environment

Rpi pi work with the Linux operating system so we will connect with the shell in all time

11.1 Working with iot

1. Start the webiopi server

```
sudo /etc/init.d/webiopi start
```

2. Start the motion server (webcam server)

```
sudo service motion start
```

3. Open the web browser in your pc and enter the rpiip address following py port number

Ex 192.168.1.3:8000 show figure 11.1 ,then enter the user name and password

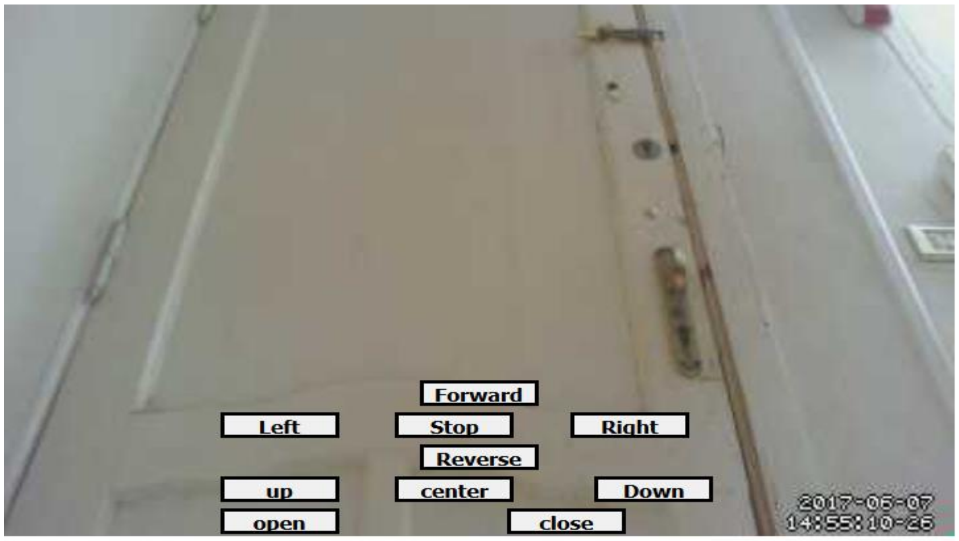


Figure 10.1

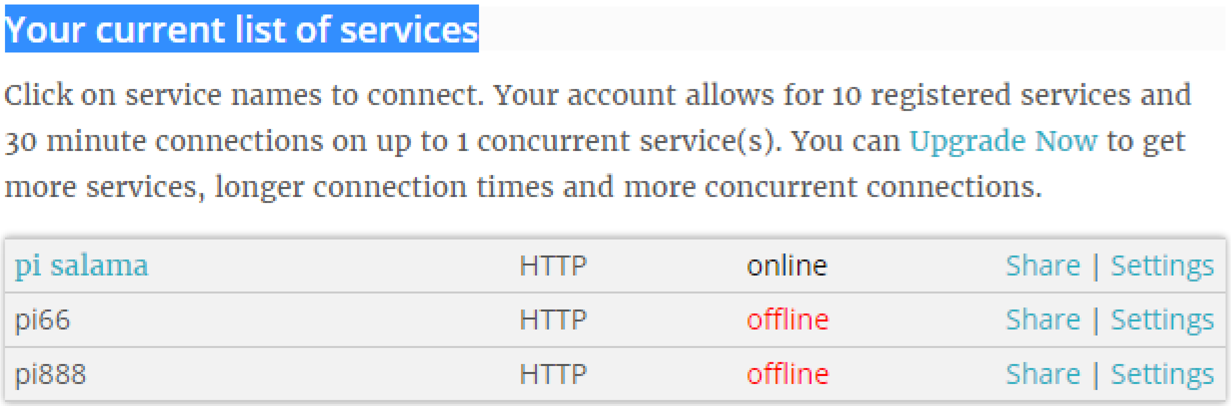
4. If needed to connect with public internet start

```
sudo /usr/bin/Weavedweb8000.sh start
```

I. Open this link from your browser https://developer.weaved.com/portal/login.php?error=NoSession
II. registered with Weaved
III. configure weaved with rpi to connect through http protocol

```
./weaved-nixinstaller_1.2.13.bin
```

IV. in the your current list of service choose your device name show figure 11.2


Your current list of services

Click on service names to connect. Your account allows for 10 registered services and 30 minute connections on up to 1 concurrent service(s). You can Upgrade Now to get more services, longer connection times and more concurrent connections.

pi salama	HTTP	online	Share | Settings
pi66	HTTP	offline	Share | Settings
pi888	HTTP	offline	Share | Settings

Figure 11.2

V. will open the iot page to control robot and you have public link able to share like this https://gqmfjkxd.p19.weaved.com/

NOTE:-
All coding and configuration about iot in APPENDIX B

11.2 VOICE COMMAND

1. Go to desktop direction by

```
Cd /home/pi/Desktop/
```

2. Run python script called voice

```
Python voice.py
```

NOTE :-
All coding and configuration about Voice-Recognitionin APPENDIX A

11.3 Computer vision

1. Go to desktop direction by

```
Cd /home/pi/Desktop/
```

2. Run python script called IMAGE

```
Python IMAGE.py
```

NOTE :-
All coding and configuration about computer vision in APPENDIX C

References

/[01]http://domoticx.com/raspberry-pi-sd-image-raspbian-linux-os
[02]http://domoticx.com/raspberry-pi-sd-image-raspbian-linux-os/
[03]http://docs.opencv.org/2.4/doc/tutorials/introduction/linux_install/linux_install.html
[04]https://pythonprogramming.net/color-filter-python-opencv-tutorial/
[05]https://pythonprogramming.net/haar-cascade-object-detection-python-opencv-tutorial/
[06]http://webiopi.trouch.com/Tutorial_Basis.html
[07] Walker, Alissa. "When Will Dubai Fix Its Burning Skyscraper Problem?".Gizmodo.Gawker Media.Retrieved 2016-01-06.
[08]https://store.fut-electronics.com/products/l298-dual-motor-driver-module-2a
https://www.google.com.eg/amp/www.instructables.com/id/Control-DC-and-stepper-motors-with-L298N-Dual-Moto/%3Famp_page%3Dtrue
[09]https://store.fut-electronics.com/products/dc-geared-motor-with-metal-gear-8-8kg-250rpm-12v-1
[10]https://www.pololu.com/product/1561
[11]https://shop.pimoroni.com/products/raspberry-pi-3
[12]https://www.amazon.com/Microsoft-T3H-00011-LifeCam-HD-3000/dp/B008ZVRAQS#featureBulletsAndDetailBullets_secondary_view_div_1496844981261
[13]https://store.fut-electronics.com/collections/robot-arm/products/2-dof-robot-arm-with-gripper

CHAPTER FOUR

PROBLEM AND FUTURE PLAN

1 Problem and solution (solved)

1.1 Rpi can't connect with xrdp server

Solution :-

Remove the xrdp , vnc4server and tightvncserver and reinstall them

```
sudo apt-get remove xrdp vnc4server tightvncserver
sudo apt-get install tightvncserver
sudo apt-get install vnc4server
sudo apt-get install xrdp
```

1.2 Opencv not able to install with cmake

Solution :-

Using cmake-gui to generate the make files .

1.3 Microphone compatible with rpi

Solution :-

Using

1. **sound card .**
2. **microphone built-in webcam.**

1.4 Webiopi server not compatible with rpi3 gpio

Solution :-

Using patch to define the 40 pin gpio

1.5 Webcam server using with iot

Solution :-

Using motion server able to sharing webcam server

1.6 Webiopi not able connect direct with public internet

Solution :-

Using Weavedweb server able to connect with local webiopi server, connect free for some time

2 Problem and solution (not solved)

- **Webcam resolution very poor Can using the LOGITECH HD PRO C920 with 15 megapixels, 1080 stream 1080p videoPrice 80$**
- **Usb Microphone compatible with rpi in high resolution Not found in Egypt**
- **Single Board ComputingRpi very slow device for this tasks**
- **Wheeled Mobile Robot not affect with environment like the humanoid robot**
- **Robot not have free moving , the wired of power supply effect in moves**
- **The voice engine not compatible with rpi 3**

3 FUTURE PLANS:-

1. **Convert the robot from the Wheeled Mobile Robot to humanoid robot.**
2. **Connect all the algorithms to gathers.**

(voice ,image recognize as automatic control and iot as manual control) .

3. **More ai in image processor to give high performance (neural network).**
4. **Using Single Board Computing with high processing power**

   (Axiomtek PICO500 Pico-ITX).

APPENDIX A

Voice-Recognition data in CD attachment with book

A. Voice.py
B. speech-rec.sh(sttgoogle cloud)

APPENDIX B

Internet of things data in CD attachment with book

A. motion.txt (config)
B. webiopi&weaved.txt (config)
C. index.html
D. script.py

APPENDIX C

Computer vision data in CD attachment with book

A. xml.py (store image & and convert from negative image to positive image)
B. IMAGE.py
C. Cascadewatch.xml
D. Testhaarcascade.py
E. haarcascade_frontalface_default.xml
F. haarcascade_eye.xml